이학 박사 학위논문

A study on high order numerical method for solving hyperbolic conservation laws

(쌍곡 보존 법칙들을 풀기 위한 고차정확도 수치 기법에 대한 연구)

2017년 2월

서울대학교 대학원

수리과학부

도성주

A study on high order numerical method for solving hyperbolic conservation laws

A dissertation
submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
to the faculty of the Graduate School of
Seoul National University

by

Seongju Do

Dissertation Director : Professor Myungjoo Kang

Department of Mathematical Sciences
Seoul National University

February 2017

Abstract

A study on high order numerical method for solving hyperbolic conservation laws

Seong-Ju Do

Department of Mathematical Sciences

The Graduate School

Seoul National University

In this thesis, we develop efficient and high order accurate numerical schemes for solving hyperbolic conservation laws such as the Euler equation and the ideal MHD(Magnetohydrodynamics) equations. The first scheme we propose is the *wavelet-based adaptive WENO method*. The Finite difference WENO scheme is one of the popular numerical schemes for application to hyperbolic conservation laws. The scheme has high order accuracy, robustness and stable property. On the other hand, the WENO scheme is computationally expensive since it performs characteristic decomposition and computes nonlinear weights for WENO interpolations. In order to overcome the drawback, we propose the adaptation technique that applies WENO differentiation for only discontinuous regions and central differentiation without characteristic decomposition for the other regions. Therefore continuous and discontinuous regions should be appropriately classified so that the adaptation method successfully works. In the wavelet-based WENO method, singularities are detected by analyzing wavelet coefficients. Such coefficients are also used to reconstruct the compressed informations.

Secondly, we propose *central-upwind schemes with modified MLP(multi-dimensional limiting process)*. This scheme decreases computational cost by simplifying the scheme itself, while the first method achieve efficiency by

skipping grid points. Generally the high-order central difference schemes for conservation laws have no Riemann solvers and characteristic decompositions but tend to smear linear discontinuities. To overcome the drawback of central-upwind schemes, we use the multi-dimensional limiting process which utilizes multi-dimensional information for slope limitation to control the oscillations across discontinuities for multi-dimensional applications.

Key words: Finite difference WENO, Wavelet Analysis, Grid adaptation, Euler equation, Ideal MHD equation, Efficient and simple numerical scheme.
Student Number: 2010–20244

Contents

CONTENTS

Chapter 1

Introduction

Hyperbolic conservation laws are written as

$$\begin{cases} \dfrac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{q}) = 0, \\ \mathbf{q}(x,0) = \mathbf{q}^0(x). \end{cases} \tag{1.0.1}$$

This equation is of practical importance with applications ranging from a variety of physical phenomena to mathematical financial modeling. Since the solution of the hyperbolic system of PDEs (1.0.1) may develop steep gradients, shock waves and contact discontinuities in finite time, it is often difficult to find the analytic solution and there is a need for robust shock-capturing schemes without creating spurious oscillations. The development of the stable, accurate and effective numerical schemes for solving the conservation laws (1.0.1) is essential for analyzing of a variety of physical phenomena as well as achieve a qualitative understanding of the behavior of their solutions. Over the past several decades, various numerical schemes have been proposed to solve the equation (1.0.1).

The most popular numerical methods for hyperbolic systems of conservation laws are the upwind schemes originated by Godunov [1], and it is extended to second-order accuracy by van Leer [2]. The weighted essentially non-oscillatory (WENO) scheme [3] is one of the most popular numerical methods of Godunov's type for hyperbolic conservation laws. WENO scheme is an improvement of the essentially non-oscillatory (ENO) scheme [4, 5],

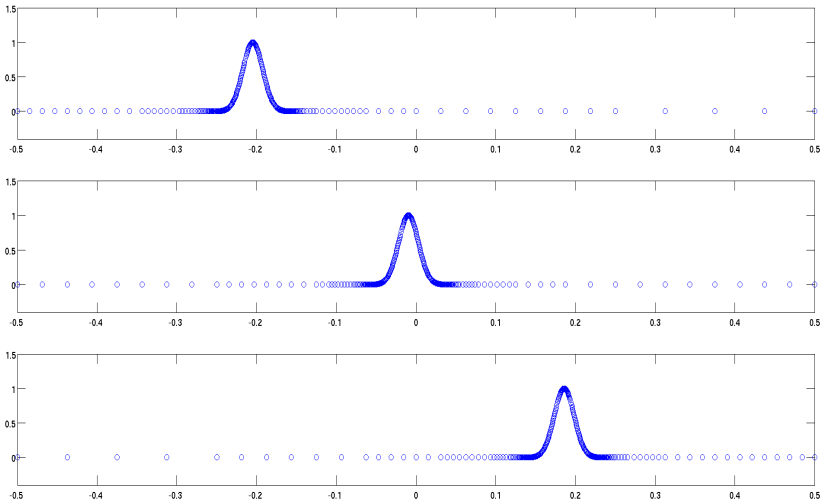
which is a uniformly high order scheme obtained by dynamically choosing the smoothest stencil for the calculations of flux derivatives. Rather than selecting only the smoothest stencil from all of the candidate stencils around the object cell, WENO assigns weights which are computed using smoothness indicators to all stencils. Therefore a higher order accuracy is obtained in the ideal case. Defining an effective smoothness indicator is an interesting topic, and several improved versions of WENO schemes have been proposed, such as WENO-M [6], WENO-Z [7] and WENO-NS [8], since the development of WENO-JS [3].

In general, the fundamental task for numerical solutions using WENO scheme is to compute interface flux values, using either cell averages (finite volume sense) or pointwise values at cell centers (finite difference sense), by applying higher order interpolations. When solving an Euler system of conservation laws using WENO schemes, flux splitting such as Lax-Friedrich splitting, combined with characteristic decompositions of a Jacobian matrix, is used in order to obtain better numerical results. Computing smoothness indicators in the characteristic spaces concerning non-linear terms at each cell makes WENO schemes being computationally expensive.

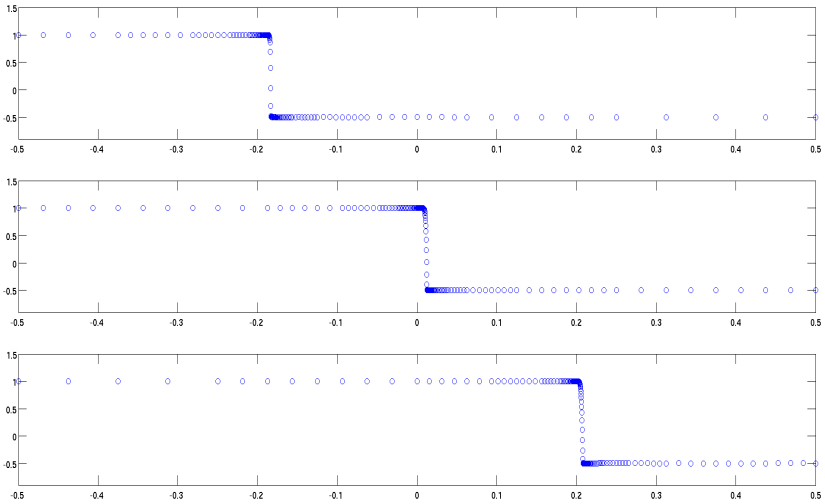
If fixed stencil finite difference or volume schemes of at least second order are applied to hyperbolic conservation laws, oscillations are produced at discontinuities. In many numerical examples, portions containing singularities where oscillations arise, are not large compared to whole computational domains. We often observe that the values contained in quite large portions of the computational domains are smooth. It is obvious that the application of the expensive WENO scheme to those areas having smooth solutions is a computational waste.

Our proposed scheme achieves adaptivity through two different aspects. One is an adaptivity of numerical grid points, and the other one is an adaptivity of differentiation methods. In Figure 1.1, a continuous Gaussian wave(Figure 1.1(a)) and a discontinuous step function(Figure 1.1 (b)) propagate towards the right direction. The continuous example of Gaussian wave shows the adaptive assignments of numerical mesh points during propagation and the discontinuous example shows that expensive high resolution schemes

are necessary only in the neighborhood of discontinuous points, and that the other points can be efficiently computed by fast fixed stencil differentiations.



(a) adaptivity of numerical grid points



(b) adaptivity of differentiation methods

Figure 1.1: Motivations of adaptivity in two aspects.

We use a lifted interpolating biorthogonal wavelet which was successfully applied by O. Vasilyev et al. [9, 10] for fluid simulations. We utilize the same multi-resolution approximation (MRA) for the assignment of numerical mesh

points in different resolution levels and for selecting different differentiation methods. The wavelet decomposition is very effective for detection of singularities. It can be seen from various figures in our numerical examples that the points of finest resolution level are automatically assigned near shock regions. By applying a hybrid of WENO differentiations at these singularities, with fixed stencil differentiations using scaling function representation at the other points, we are able to obtain pleasing numerical results. We call this numerical method *wavelet-based adaptive WENO*.

Another method to solve hyperbolic conservation laws is the Lax-Friedrichs (LxF) central difference scheme introduced by Lax and Fridrichs in [11, 12], in which no Riemann solvers and characteristic decomposition are involved. It has the advantage of simplicity compared with the first-order upwind scheme of Godunov [1]. Unfortunately, the LxF scheme, however, yields large numerical dissipations, which leads to a poor resolution of shock discontinuities and rarefaction waves. In order to reduce the numerical dissipation, Nessyahu and Tadmor proposed a second-order extension of the LxF scheme (refered to NT scheme) in [13], which is based on the staggered form of the Lax-Friedrichs scheme. The NT scheme replaces the first-order piecewise constant solution with van Leer's MUSCL-type piecewise-linear one to construct second-order approximation, while avoiding oscillations at discontinuities and achieving a sharp and accurate shock capturing. The NT scheme retains the simplicity of the Riemann-free LxF framework and avoids the disadvantage of excessive first order dissipation of the LxF scheme. In the year 2000, Kurganov and Tadmor (CU-KT) [14] proposed modifications to the NT scheme with a smaller amount of numerical viscosity than that of the original NT scheme. The second-order KT schemes with a semidiscrete formulation were based on integration over Riemann fans of variable sizes and used more precise information for the local speeds of propagation. Extensions to multidimensional problems were introduced in [15].

Later generalization of the CU-KT schemes was proposed by Kurganov, Noelle and Petrova [16] utilized one-sided local speeds of propagation(referred to CU-KNP scheme). The numerical flux of the KT scheme uses only the maximum wave speed but the CU-KNP scheme uses both maximal and

minimum wave speeds to reduce the numerical viscosity. Higher than second order schemes presented by Liu and Osher [17] are based on the non-oscillatory third-order reconstruction with staggered evolution of the reconstructed cell averages[18]. High-order essentially non-oscillatory (ENO) [19] and weighted ENO (WENO) [3, 20] reconstructions were combined with central schemes by Bianco, Puppo and Russo [21]. Modifications of central schemes of Bianco, Puppo and Russo [21] were introduced in [22], which is a central weighted non-oscillatory (CWENO) reconstruction. Other central schemes with WENO are presented in [23, 24] and extensions to multi-dimensional problems can be found in [25, 26]. The higher order reconstructions of central schemes have smaller numerical dissipations and yield a higher resolution of shocks, rarefactions. Furthermore, other spontaneous evolution of large gradient phenomena almost as sharply as comparable higher order upwind schemes.

Most of the numerical schemes for hyperbolic conservation laws have been developed on one-dimensional flow physics so that two-dimensional flows are extended in a straightforward way by dimensional splitting method. Although this approach allows the rigorous analysis of numerical schemes, using one-dimensional limiters is not successful to achieve a good shock resolution if the shock is located in direction diagonal to the computational grid. These numerical schemes lead to insufficient or excessive numerical dissipation due to the essential limitations of the accurate and efficient calculations for multi-dimensional flows. During the last two decades there were activities to control numerical oscillation and to design for multi-dimensional limiter functions [27, 28, 29]. As successive studies, Kim and Kim extended the one-dimensional monotonic condition to two-dimensional flow and proposed multi-dimensional limiting process (MLP) for the two-dimensional compressible Euler [30]. They proposed a multi-dimensional limiting process (MLP) in the same article which uses multi-dimensional informations for slope limitation to control the oscillations across discontinuities for multi-dimensional applications. Later, improved MLP limiters have been devised [31, 32, 33, 34] and they could be efficiently implemented in three-dimensional space. The main focus of the MLP methods is to eliminate excessive numerical dissipa-

tions and upgrade solution accuracy by predicting the physical distributions of flow variables in multi-space dimensions. We propose *CU-MLP(Central-Upwind method with MLP)* to improve the performance of CU-KNP, maintaining the advantages of CU-KNP mentioned above.

The outline of this thesis is as follows : In Section 2 we define hyperbolic conservation laws to be solved numerically and summarize eigen-structures of the Euler equation and the ideal MHD equation. The well-known techniques to impose divergence-free constraint in ideal MHD equation are briefly described. The finite difference WENO methods and basic wavelet theories are reviewed in order to explain the key concept of wavelet-based adaptive WENO method in Section 3. The central upwind with MLP method is introduced in Section 4. Finally, conclusions are given in Section 5.

Chapter 2

Governing Equations

In this chapter we introduce the concepts of a hyperbolic conservation law. The Euler equation and Ideal MHD(Magnetohydrodynamics)equation, which are popular applications of a hyperbolic conservation law, are described. Furthermore, the eigen-structure of these equations are derived so that characteristic decomposition method can be applied.

2.1 Hyperbolic Conservation Laws

A physical quantity $q(\mathbf{x}, t)$ is said to be *conserved* if the amount of q in some bounded region $\Omega \subset \mathbb{R}^n$ can only be modified in time by the flux $\mathbf{F}(q)$ passing through the boundary of Ω :

$$\begin{aligned} \frac{\partial}{\partial t}\left(\int_{\Omega} q\, dV\right) &= \text{net flux across } \partial\Omega \\ &= -\int_{\partial\Omega} \mathbf{F}\cdot\mathbf{n}\, dS, \end{aligned} \tag{2.1.1}$$

where $\mathbf{n}$ is the outward unit normal vector of $\partial\Omega$. Divergence theorem can be applied to the last expression in Eq (2.1.1) under the assumption of smoothness on q :

$$\int_{\Omega}\left(\frac{\partial q}{\partial t} + \nabla\cdot\mathbf{F}\right) dV = 0. \tag{2.1.2}$$

The Eq (2.1.2) is called the *integral form* of a conservation law. Since this integral must be zero for arbitrary region Ω, it follows that the integrand must be identically zero. This gives the differential equation

$$\frac{\partial q}{\partial t} + \nabla \cdot \mathbf{F} = 0. \tag{2.1.3}$$

The Eq (2.1.3) is called the *differential form* of a conservation law.

Most of PDEs(partial differential equations) can be classified as *elliptic*, *parabolic* and *hyperbolic*. We will focus on only hyperbolic equations. The classical hyperbolic PDE is the following *wave equation* :

$$\frac{\partial^2 q}{\partial t^2} - \frac{\partial^2 q}{\partial x^2} = 0 \tag{2.1.4}$$

$$q(x, 0) = q_0(x). \tag{2.1.5}$$

The PDE propagates the initial condition $q_0(x)$ with the wave speed ± 1. For the hyperbolic case, unlike elliptic and parabolic PDEs, physical information travels at finite speeds. Since the propagation of waves occurs in many important physical applications, it is crucial to develop numerical method for solving the wave equation and its generalizations. In order to consider the general situation, rigorous meaning of hyperbolic conservation laws is introduced [35]. The dimensions of variables in (2.1.3) are defined as follows :

$$\mathbf{q} \in \mathbb{R}^m, \quad \mathbf{F} \in \mathbb{R}^{m \times n} \quad \text{and} \quad \mathbf{v} \in \mathbb{R}^n, \tag{2.1.6}$$

where m is the number of unknowns and n is the number of spatial dimensions. In other words, the PDE (2.1.3) is extended to a system of PDEs to cover more general problems.

Definition 2.1.1 (Hyperbolicity)**.** Eq (2.1.3) is *hyperbolic* if, for any $\mathbf{v} \in \mathbb{R}^n$ such that $|\mathbf{v}| = 1$, the Jacobian matrix $\frac{\partial}{\partial q}(\mathbf{F} \cdot \mathbf{v})$ has real eigenvalues with a complete family of right eigenvectors. Furthermore, the system is said to be *strictly hyperbolic* if the eigenvalues are distinct.

As it will be seen in subsequent subsections, the eigenvalues of the flux Jacobian are directly related to the *wave speeds* of solutions to the conservation

law. Since the eigenvalues are required to be real, physical dynamics governed by hyperbolic conservation laws necessarily involve information propagating at finite speeds.

In our work, we focus on the hyperbolic conservation laws represented as

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial}{\partial x}\mathbf{F}(\mathbf{q}) + \frac{\partial}{\partial y}\mathbf{G}(\mathbf{q}) + \frac{\partial}{\partial z}\mathbf{H}(\mathbf{q}) = \mathbf{0}. \tag{2.1.7}$$

2.2 Euler equation

2.2.1 Model equation

The 3-D unsteady Euler equations of inviscid flow, a system of integral conservation equations for mass, momentum and energy, can be expressed as Eq (2.1.7) with

$$\mathbf{q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (E+p)u \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (E+p)v \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (E+p)w \end{pmatrix}, \tag{2.2.1}$$

where ρ, u, v, w, p and E are the density, x-directional velocity, y-directional velocity, z-directional velocity, thermal pressure and total energy, respectively. Each physical quantities are connected by specifying the equation of state :

$$p = (\gamma - 1)\left(E - \frac{\rho(u^2 + v^2 + w^2)}{2}\right), \tag{2.2.2}$$

with a specific heat ratio γ.

For 2-D Euler equations, z-directional velocity w is treated a constant field. Then the fourth PDE of system (2.1.7) is reduced to the first PDE(namely, the density conservation). Therefore 2-D Euler equation can be formed by just dropping fourth rows of every vectors and $\mathbf{H}$ itself in Eq (2.2.1). 1-D

case can be handled in the same manner.

2.2.2 Eigen-structure

The main goal of this subsection is to derive the explicit expression of eigenvalues and eigenvectors for the Jacobian matrices $\frac{\partial \mathbf{F}}{\partial \mathbf{q}}$, $\frac{\partial \mathbf{G}}{\partial \mathbf{q}}$ and $\frac{\partial \mathbf{H}}{\partial \mathbf{q}}$. It is enough to obtain of eigen-structures of $\frac{\partial \mathbf{F}}{\partial \mathbf{q}}$ by the following observation.

Proposition 2.2.1. *After switching some variables, Jacobian matrices* $\frac{\partial \mathbf{F}}{\partial \mathbf{q}}$, $\frac{\partial \mathbf{G}}{\partial \mathbf{q}}$ *and* $\frac{\partial \mathbf{H}}{\partial \mathbf{q}}$ *are similar each other.*

Proof. It is sufficient to show that $\frac{\partial \mathbf{F}}{\partial \mathbf{q}}$ and $\frac{\partial \mathbf{G}}{\partial \mathbf{q}}$ are similar after switching some variables. Define an operator $\mathcal{A}$ as the switching operator of variables u and v. Also define an elementary matrix interchanging the second and third rows, namely

$$E = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2.2.3}$$

It is trivial that $\mathcal{A}$ preserves a product of matrices. Then, it follows that

$$\mathbf{G} = \mathcal{A}(E\mathbf{F}) = \mathcal{A}(E)\mathcal{A}(\mathbf{F}) = E\mathcal{A}(\mathbf{F}). \tag{2.2.4}$$

By the linearity of differential operator,

$$\frac{\partial \mathbf{G}}{\partial \mathbf{q}} = \frac{\partial (E\mathcal{A}(\mathbf{F}))}{\partial \mathbf{q}} = E\frac{\partial (\mathcal{A}(\mathbf{F}))}{\partial \mathbf{q}}. \tag{2.2.5}$$

And we introduce an auxiliary vector $\widehat{\mathbf{q}}$ as

$$\widehat{\mathbf{q}} = \begin{pmatrix} \widehat{q}_1 \\ \widehat{q}_2 \\ \widehat{q}_3 \\ \widehat{q}_4 \\ \widehat{q}_5 \end{pmatrix} = \begin{pmatrix} q_1 \\ q_3 \\ q_2 \\ q_4 \\ q_5 \end{pmatrix} = \mathcal{A}(\mathbf{q}). \tag{2.2.6}$$

Using the chain rule, we obtain that

$$\frac{\partial \mathcal{A}(\mathbf{F})}{\partial \mathbf{q}} = \frac{\partial \mathbf{F}(\widehat{\mathbf{q}})}{\partial \mathbf{q}} \tag{2.2.7}$$

$$= \frac{\partial \mathbf{F}(\widehat{\mathbf{q}})}{\partial \widehat{\mathbf{q}}} \frac{\partial \widehat{\mathbf{q}}}{\partial \mathbf{q}} \tag{2.2.8}$$

$$= \mathcal{A}\left(\frac{\partial \mathbf{F}}{\partial \mathbf{q}}\right) E. \tag{2.2.9}$$

Putting the above discussion together, we have

$$\frac{\partial \mathbf{G}}{\partial \mathbf{q}} = E\mathcal{A}\left(\frac{\partial \mathbf{F}}{\partial \mathbf{q}}\right) E. \tag{2.2.10}$$

□

The eigenvalues, right eigenvector and left eigenvector of $\frac{\partial \mathbf{F}}{\partial \mathbf{q}}$ are given as

$$\lambda_1 = u - c,\ \lambda_2 = \lambda_3 = \lambda_4 = u,\ \lambda_5 = u + c, \tag{2.2.11}$$

$$r_1 = \begin{pmatrix} 1 \\ u-c \\ v \\ w \\ H - uc \end{pmatrix}, r_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ v \end{pmatrix}, r_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ w \end{pmatrix}, r_4 = \begin{pmatrix} 1 \\ u \\ v \\ w \\ K \end{pmatrix}, r_5 = \begin{pmatrix} 1 \\ u+c \\ v \\ w \\ H + uc \end{pmatrix}, \tag{2.2.12}$$

$$l_1 = \frac{\gamma - 1}{2c^2} \begin{pmatrix} K + uc/(\gamma - 1) \\ -u - c/(\gamma - 1) \\ -v \\ -w \\ 1 \end{pmatrix}, \tag{2.2.13}$$

$$l_2 = \begin{pmatrix} -v \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad l_3 = \begin{pmatrix} -w \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \tag{2.2.14}$$

$$l_4 = \frac{\gamma - 1}{c^2} \begin{pmatrix} c^2/(\gamma - 1) - K \\ u \\ v \\ w \\ -1 \end{pmatrix}, \tag{2.2.15}$$

$$l_5 = \frac{\gamma - 1}{2c^2} \begin{pmatrix} K - uc/(\gamma - 1) \\ -u + c/(\gamma - 1) \\ -v \\ -w \\ 1 \end{pmatrix}, \tag{2.2.16}$$

where a speed of sound c, entropy H and kinetic energy K are defined as $c = \sqrt{\frac{p\gamma}{\rho}}$, $H = \frac{E+p}{\rho}$, $K = \frac{u^2+v^2+w^2}{2}$.

By the hyperbolicity of Euler equation, therefore, we can diagonalize the Jacobian matrix of $\frac{\partial \mathbf{F}}{\partial \mathbf{q}}$ as follows :

$$\frac{\partial \mathbf{F}}{\partial \mathbf{q}} = RDL, \quad LR = I_{5\times 5} \tag{2.2.17}$$

where

$$R = \left(\begin{array}{c|c|c|c|c} r_1 & r_2 & r_3 & r_4 & r_5 \end{array} \right), \quad L = \left(\begin{array}{c} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \end{array} \right), \quad D = diag(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5). \tag{2.2.18}$$

The following simple algebras derive the eigen-structures of the rest Jacobian matrices. The above discussion implies that

$$\frac{\partial \mathbf{F}}{\partial \mathbf{q}} = RDL. \tag{2.2.19}$$

By applying the operator $\mathcal{A}$ on both sides,

$$\mathcal{A}\left(\frac{\partial \mathbf{F}}{\partial \mathbf{q}}\right) = \mathcal{A}(R)\mathcal{A}(D)\mathcal{A}(L). \tag{2.2.20}$$

Multiplying the elementary matrix E which interchanges the second and third rows, we have

$$E\mathcal{A}\left(\frac{\partial \mathbf{F}}{\partial \mathbf{q}}\right)E = E\mathcal{A}(R)\mathcal{A}(D)\mathcal{A}(L)E. \tag{2.2.21}$$

Therefore,

$$\frac{\partial \mathbf{G}}{\partial \mathbf{q}} = \underbrace{\mathcal{A}(ER)}_{\widetilde{R}}\underbrace{\mathcal{A}(D)}_{\widetilde{D}}\underbrace{\mathcal{A}(LE)}_{\widetilde{L}} \tag{2.2.22}$$

The Jacobian matrix $\frac{\partial \mathbf{H}}{\partial \mathbf{q}}$ can be handled in the similar way. The observation that the eigen-structure of all Jacobian matrices of the Euler equation has similar structures makes the code implementation much easier.

2.3 Ideal MHD equation

2.3.1 Model equation

The ideal magnetohydrodynamics(MHD) equations are given by

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho\mathbf{u}\\ E\\ \mathbf{B}\end{pmatrix}+\nabla\cdot\begin{pmatrix}\rho\mathbf{u}\\ \rho\mathbf{u}\otimes\mathbf{u}+\widetilde{p}I_{3\times3}-\mathbf{B}\otimes\mathbf{B}\\ \mathbf{u}(E+\widetilde{p})-\mathbf{B}(\mathbf{B}\cdot\mathbf{u})\\ \mathbf{u}\otimes\mathbf{B}-\mathbf{B}\otimes\mathbf{u}\end{pmatrix}=\mathbf{0},\qquad(2.3.1)$$

$$\nabla\cdot\mathbf{B}=0,$$

where

$$\widetilde{p}=p+\frac{1}{2}\left\|\mathbf{B}\right\|^2,\qquad(2.3.2)$$

$$p=(\gamma-1)\left(E-\frac{1}{2}\rho\left\|\mathbf{u}\right\|^2-\frac{1}{2}\left\|\mathbf{B}\right\|^2\right).\qquad(2.3.3)$$

In these equations, ρ is the density, $\mathbf{u}=(u,v,w)$ is the velocity field, E is the total energy, $\mathbf{B}=(B_1,B_2,B_3)$ is the magnetic field, p is the thermal pressure, $\frac{1}{2}\left\|\mathbf{B}\right\|^2$ is the magnetic pressure, $\widetilde{p}$ is the total pressure, and γ is the specific heat ratio. These equation describe the dynamics of an elcetrically conducting fluid and are a combination of gas dynamics and Maxwell's equations. The Eq (2.3.1) can be expressed in the form Eq (2.1.7) :

$$\mathbf{q}=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ E\\ B_1\\ B_2\\ B_3\end{pmatrix},\ \mathbf{F}=\begin{pmatrix}\rho u\\ \rho u^2+\widetilde{p}-B_1^2\\ \rho uv-B_1B_2\\ \rho uw-B_1B_3\\ (E+\widetilde{p})u-(\mathbf{B}\cdot\mathbf{u})B_1\\ 0\\ uB_2-B_1v\\ uB_3-B_1w\end{pmatrix}$$

$$\mathbf{G} = \begin{pmatrix} \rho v \\ \rho uv - B_2B_1 \\ \rho v^2 + \widetilde{p} - B_2^2 \\ \rho vw - B_2B_3 \\ (E+\widetilde{p})v - (\mathbf{B}\cdot\mathbf{u})B_2 \\ vB_1 - B_2u \\ 0 \\ vB_3 - B_2w \end{pmatrix}, \ \mathbf{H} = \begin{pmatrix} \rho w \\ \rho uw - B_3B_1 \\ \rho vw - B_3B_2 \\ \rho w^2 + \widetilde{p} - B_3^2 \\ (E+\widetilde{p})w - (\mathbf{B}\cdot\mathbf{u})B_3 \\ wB_1 - B_3u \\ wB_2 - B_3v \\ 0 \end{pmatrix}$$

It is worth noting that the ideal MHD equation with $\mathbf{B} = \mathbf{0}$ is reduced to the Euler equation.

2.3.2 Eigen-Structure

In this subsection, we process the similar works done in the section 2.2.2. We derive the flux Jacobian for the ideal MHD equations and compute its eigen-decomposition. Notice that in each direction there are only seven non-trivial equations and one for which the time derivative is identically zero. Because of this, the flux Jacobian will be an 8×8 matrix, but with zeros placed in the row and column that corresponds to the trivial equation.

It is enough to introduce eigen-decomposition of $\frac{\partial \mathbf{F}}{\partial \mathbf{q}}$ with the aid of the following relations :

$$\mathbf{G} = E\mathcal{A}(F) = \mathcal{A}(EF), \tag{2.3.4}$$

where $\mathcal{A}$ is an operator switching $u \leftrightarrow v$, $B_1 \leftrightarrow B_2$, and E is the 8×8 elementary matrix that interchanges $2nd\ row \leftrightarrow 3rd\ row$ and $6th\ row \leftrightarrow 7th\ row$.

The eigenvalues of $\frac{\partial \mathbf{F}}{\partial \mathbf{q}}$ are given by

$$\begin{aligned}
\lambda_{1,8} &= u \mp c_f, && \text{(Fast magnetic waves)} \\
\lambda_{2,7} &= u \mp c_a, && \text{(Alfven waves)} \\
\lambda_{3,6} &= u \mp c_s, && \text{(Slow magnetic waves)} \\
\lambda_4 &= u, && \text{(Entropy wave)} \\
\lambda_5 &= u, && \text{(Divergence wave)}
\end{aligned}$$

where

$$a = \sqrt{\frac{\gamma p}{\rho}}, \tag{2.3.5}$$

$$c_a = \sqrt{\frac{B_1^2}{\rho}}, \tag{2.3.6}$$

$$c_{s,f} = \left[\frac{1}{2} \left(a^2 + \frac{\|\mathbf{B}\|^2}{\rho} \mp \sqrt{\left(a^2 + \frac{\|\mathbf{B}\|^2}{\rho} \right)^2 - 4a^2 \frac{B_1^2}{\rho}} \right) \right]^{1/2}. \tag{2.3.7}$$

Note that the eigenvalues satisfy the inequalities

$$\lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \lambda_4 \leq \lambda_5 \leq \lambda_6 \leq \lambda_7 \leq \lambda_8 \tag{2.3.8}$$

We now aim to find a decomposition of these matrices such that

$$\frac{\partial \mathbf{F}}{\partial \mathbf{q}} = RDL, \tag{2.3.9}$$

where

$$R = \left(\begin{array}{c|c|c|c|c|c|c} r_1 & r_2 & r_3 & r_4 & r_5 & r_6 & r_7 \end{array} \right), \quad L = \begin{pmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \\ l_7 \end{pmatrix}, \tag{2.3.10}$$

$$D = diag(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_6, \lambda_7, \lambda_8) \tag{2.3.11}$$

For the simplicity of notation, we introduce the temporary variables [36]

$$\mu_f^{\mp} = \frac{\alpha_f c_f^2}{\gamma - 1} \mp \alpha_f c_f u \pm \alpha_s c_a \beta_1 (\beta_2 v + \beta_3 w) + \frac{\gamma - 2}{\gamma - 1} \alpha_f (c_f^2 - a^2), \tag{2.3.12}$$

$$\mu_s^{\mp} = \frac{\alpha_s c_s^2}{\gamma - 1} \mp \alpha_s c_s u \mp \alpha_f a \beta_1 (\beta_2 v + \beta_3 w) + \frac{\gamma - 2}{\gamma - 1} \alpha_s (c_s^2 - a^2), \tag{2.3.13}$$

$$\alpha_f = \begin{cases} 1 & \text{if} \quad B_2 = B_3 = 0 \quad \text{and} \quad c_a^2 = a^2 \\ \sqrt{(c_f^2 - c_a^2)/(c_f^2 - c_s^2)} & \text{otherwise} \end{cases} \tag{2.3.14}$$

$$\alpha_s = \begin{cases} 1 & \text{if} \quad B_2 = B_3 = 0 \quad \text{and} \quad c_a^2 = a^2 \\ \sqrt{(c_f^2 - a^2)/(c_f^2 - c_s^2)} & \text{otherwise} \end{cases} \tag{2.3.15}$$

$$\beta_1 = \text{sign}(B_1), \tag{2.3.16}$$

$$\beta_2 = \begin{cases} 1/\sqrt{2} & \text{if} \quad B_2 = B_3 = 0 \\ B_2/\sqrt{B_2^2 + B_3^2} & \text{otherwise} \end{cases} \tag{2.3.17}$$

$$\beta_3 = \begin{cases} 1/\sqrt{2} & \text{if} \quad B_2 = B_3 = 0 \\ B_3/\sqrt{B_2^2 + B_3^2} & \text{otherwise} \end{cases} \tag{2.3.18}$$

$$\theta_1 = \frac{1}{2} \left[\alpha_f^2 a^2 \left(c_f^2 - \frac{\gamma - 2}{\gamma - 1} a^2 \right) + \alpha_s^2 c_f^2 \left(c_s^2 - \frac{\gamma - 2}{\gamma - 1} a^2 \right) \right]^{-1}, \tag{2.3.19}$$

$$\theta_2 = \frac{1}{2} \left[\alpha_f^2 c_f a \beta_1 + \alpha_s^2 c_s c_a \beta_1 . \right]^{-1}. \tag{2.3.20}$$

The right and left eigenvectors are expressed by

$$r_{1,7} = \begin{pmatrix} \alpha_f \\ \alpha_f(u \mp c_f) \\ \alpha_f v \pm \alpha_s \beta_1 \beta_2 c_a \\ \alpha_f w \pm \alpha_s \beta_1 \beta_3 c_a \\ \frac{1}{2}\alpha_f \|\mathbf{u}\|^2 + \mu_f^{\mp} \\ 0 \\ \frac{1}{\sqrt{\rho}}\alpha_s \beta_2 c_f \\ \frac{1}{\sqrt{\rho}}\alpha_s \beta_3 c_f \end{pmatrix}, \tag{2.3.21}$$

$$r_{2,6} = \begin{pmatrix} 0 \\ 0 \\ \pm\beta_1\beta_2 \\ \mp\beta_1\beta_2 \\ \pm(\beta_3 v - \beta_2 w)\beta_1 \\ 0 \\ \dfrac{1}{\sqrt{\rho}}\beta_3 \\ -\dfrac{1}{\sqrt{\rho}}\beta_2 \end{pmatrix}, \quad r_4 = \begin{pmatrix} 1 \\ u \\ v \\ w \\ \frac{1}{2}\|\mathbf{u}\|^2 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{2.3.22}$$

$$r_{3,5} = \begin{pmatrix} \alpha_s \\ \alpha_s(u \mp c_s) \\ \alpha_s v \mp \alpha_f \beta_1 \beta_2 a \\ \alpha_s w - \mp\alpha_f \beta_1 \beta_3 a \\ \frac{1}{2}\alpha_s \|\mathbf{u}\|^2 + \mu_s^{\mp} \\ 0 \\ -\frac{1}{c_f\sqrt{\rho}}\alpha_f \beta_2 a^2 \\ -\frac{1}{c_f\sqrt{\rho}}\alpha_f \beta_3 a^2 \end{pmatrix}. \tag{2.3.23}$$

The left eigenvectors are written explicitly as

$$l_{1,7} = \begin{pmatrix} \frac{1}{2}\theta_1\alpha_f a^2 \|\mathbf{u}\|^2 \pm \theta_2[\alpha_f a u\beta_1 - \alpha_s c_s(\beta_2 v + \beta_3 w)] \\ -\theta_1\alpha_f a^2 u \mp \theta_2\alpha_f a\beta_1 \\ -\theta_1\alpha_f a^2 v \pm \theta_2\alpha_s c_s\beta_2 \\ -\theta_1\alpha_f a^2 w \pm \theta_2\alpha_s c_s\beta_3 \\ \theta_1\alpha_f a^2 \\ 0 \\ \theta_1\sqrt{\rho}\alpha_s\beta_2 c_f\left(c_s^2 - \frac{\gamma-2}{\gamma-1}a^2\right) \\ \theta_1\sqrt{\rho}\alpha_s\beta_3 c_f\left(c_s^2 - \frac{\gamma-2}{\gamma-1}a^2\right) \end{pmatrix} \tag{2.3.24}$$

$$l_{2,6} = \begin{pmatrix} \mp\frac{1}{2}\beta_1(\beta_3 v - \beta_2 w) \\ 0 \\ \pm\frac{1}{2}\beta_1\beta_3 \\ \pm\frac{1}{2}\beta_1\beta_2 \\ 0 \\ 0 \\ \frac{1}{2}\sqrt{\rho}\beta_3 \\ -\frac{1}{2}\sqrt{\rho}\beta_2 \end{pmatrix}, \quad l_4 = \begin{pmatrix} 1 - \theta_1\left(\alpha_f^2 a^2 + \alpha_s^2 c_f^2 \|\mathbf{u}\|^2\right) \\ 2\theta_1\left(\alpha_f^2 a^2 + \alpha_s^2 c_f^2\right) u \\ 2\theta_1\left(\alpha_f^2 a^2 + \alpha_s^2 c_f^2\right) v \\ 2\theta_1\left(\alpha_f^2 a^2 + \alpha_s^2 c_f^2\right) w \\ -2\theta_1\left(\alpha_f^2 a^2 + \alpha_s^2 c_f^2\right) \\ 0 \\ 2\theta_1\sqrt{\rho}\alpha_f\alpha_s\beta_2 c_f\left(c_f^2 - c_s^2\right) \\ 2\theta_1\sqrt{\rho}\alpha_f\alpha_s\beta_3 c_f\left(c_f^2 - c_s^2\right) \end{pmatrix} \tag{2.3.25}$$

$$l_{3,5} = \begin{pmatrix} \frac{1}{2}\theta_1\alpha_s c_f^2 \|\mathbf{u}\|^2 \pm \theta_2[\alpha_s c_a u\beta_1 + \alpha_f c_f(\beta_2 v + \beta_3 w)] \\ -\theta_1\alpha_s c_f^2 u \mp \theta_2\alpha_s c_a\beta_1 \\ -\theta_1\alpha_s c_f^2 v \mp \theta_2\alpha_f c_f\beta_2 \\ -\theta_1\alpha_s c_f^2 w \mp \theta_2\alpha_f c_f\beta_3 \\ \theta_1\alpha_s c_f^2 \\ 0 \\ \theta_1\sqrt{\rho}\alpha_f\beta_2 c_f\left(c_f^2 - \frac{\gamma-2}{\gamma-1}a^2\right) \\ -\theta_1\sqrt{\rho}\alpha_f\beta_3 c_f\left(c_f^2 - \frac{\gamma-2}{\gamma-1}a^2\right) \end{pmatrix} \tag{2.3.26}$$

Note that the condition, in Eq (2.3.14), $B_2 = B_3 = 0$ and $c_a^2 = a^2$ implies

that

$$c_f = c_2 = c_a = a.$$

Hence, conditional sentences "if" are suitably introduced so as to avoid dividing by zeros.

2.4 The $\nabla \cdot \mathbf{B} = 0$ Constraint in MHD Codes

Our goal is to develop MHD codes that successfully solve many problems involving all kinds of discontinuities. Still, there are unresolved arguments related to how one should maintain the divergence-free property of the magnetic field in multidimensional MHD simulations. This constraint is automatically satisfied in one-dimensional simulations since $\frac{\partial}{\partial y} = \frac{\partial}{\partial z} = 0$ and consequently $\frac{\partial B_1}{\partial x} = 0$, namely $B_1 =$ constant, but many numerical schemes do not guarantee $\nabla \cdot \mathbf{B} = 0$ in multidimensional cases.

The paper [37] summarizes and compares the three approaches, *8-waves formulation*, *constraint transport(CT)* and *projection scheme*, to impose divergence-free constraint to a base numerical scheme. In the paper [38], furthermore, a *divergence cleaning* technique that introduces an additional scalar field to propagate the divergence errors outside of the computational domain is devised. Since the divergence cleaning method is suitable to the framework of the grid adaptation, the method is adopted to our scheme appearing in the chapter 3. The CT method is chosen to implement the simulation with the numerical method that is introduced in chapter 4.

2.4.1 Constraints Transport Method

For sake of simplicity, the discrete equations will be shown for a two-dimensional uniform Cartesian grid. The key idea of the CT method [39] is to represent the magnetic flux as the curl of magnetic potential $\mathbf{A} = -\mathbf{u} \times \mathbf{B}$. Then the

PDEs according to the magnetic part of Eq (2.3.1) is rewritten as

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{A} = \mathbf{0}. \tag{2.4.1}$$

Taking a divergence on the both sides, we have

$$\frac{\partial (\nabla \cdot \mathbf{B})}{\partial t} = 0$$

This implies that magnetic divergence always keeps the constant state. In the light of numerical schemes, this fact can be represented as

$$\nabla \cdot \mathbf{B}^n = 0 \quad \Rightarrow \quad \nabla \cdot \mathbf{B}^{n+1} = 0. \tag{2.4.2}$$

That is, the divergence of magnetic fields must be zero provided the initial magnetic divergence is zero. If the magnetic field $\mathbf{B}$ is sufficiently smooth, Eq (2.4.2) should hold for any numerical scheme. It is also crucial to devise a numerical scheme having the property Eq (2.4.2) in the case that the magnetic field $\mathbf{B}$ is discontinuous. It turns out that *flux CT* method is numerically stable and gives reasonable solutions [37]. Thus we first describe this method.

Note that $\mathbf{A}$ is independent of a spatial variable z in a two-dimensional case. The Eq (2.4.1) is written as a component-wisely :

$$\begin{aligned} \frac{\partial B_1}{\partial t} + \frac{\partial \Omega}{\partial y} &= 0, \\ \frac{\partial B_2}{\partial t} - \frac{\partial \Omega}{\partial x} &= 0, \end{aligned} \tag{2.4.3}$$

where $\Omega = vB_1 - uB_2$ is the $z-$component of potential $\mathbf{A}$. The main idea of CT method is very simple. Firstly, we start with the semi-discrete Euler

update of the Eq (2.4.3) :

$$\begin{aligned}(B_1)_{i,j}^{n+1} &= (B_1)_{i,j}^{n} - \Delta t \left(\frac{\partial \Omega}{\partial y}\right)_{i,j}, \\ (B_2)_{i,j}^{n+1} &= (B_2)_{i,j}^{n} + \Delta t \left(\frac{\partial \Omega}{\partial x}\right)_{i,j}.\end{aligned} \tag{2.4.4}$$

Taking the divergence operator on the both sides,

$$\left(\frac{\partial B_1}{\partial x}\right)_{i,j}^{n+1} + \left(\frac{\partial B_2}{\partial y}\right)_{i,j}^{n+1} = \left(\frac{\partial B_1}{\partial x}\right)_{i,j}^{n} + \left(\frac{\partial B_2}{\partial y}\right)_{i,j}^{n} + \Delta t \left(- \left(\frac{\partial^2 \Omega}{\partial x \partial y}\right)_{i,j} + \left(\frac{\partial^2 \Omega}{\partial y \partial x}\right)_{i,j} \right). \tag{2.4.5}$$

If we choose some consistent approximation to the operator $\dfrac{\partial^2}{\partial x \partial y}$, the property (2.4.2) is satisfied.

In flux CT method, the magnetic potential field is assigned at a cell face, thus the Eq (2.4.3) is written as

$$(B_1)_{i,j}^{n+1} = (B_1)_{i,j}^{n} - \Delta t \frac{\Omega_{i,j+1/2} - \Omega_{i,j-1/2}}{\Delta y}, \tag{2.4.6}$$

$$(B_2)_{i,j}^{n+1} = (B_2)_{i,j}^{n} + \Delta t \frac{\Omega_{i+1/2,j} - \Omega_{i-1/2,j}}{\Delta x}. \tag{2.4.7}$$

Then, we define Ω as follows :

$$\begin{aligned}\Omega_{i+1/2,j} = -\frac{1}{8}\big(&2f_{i+1/2,j} + f_{i+1/2,j+1} + f_{i+1/2,j-1} \\ &- g_{i,j+1/2} - g_{i+1,j+1/2} - g_{i,j-1/2} - g_{i+1,j-1/2}\big), \\ \Omega_{i,j+1/2} = -\frac{1}{8}\big(&f_{i+1/2,j} + f_{i+1/2,j+1} + f_{i-1/2,j} + f_{i-1/2,j+1} \\ &- 2g_{i,j+1/2} - g_{i+1,j+1/2} - g_{i-1,j+1/2}\big),\end{aligned} \tag{2.4.8}$$

where f and g are fluxes of arbitrary base scheme corresponding to B_2 and B_1, respectively. It is easy to verify that the following cell corner centered

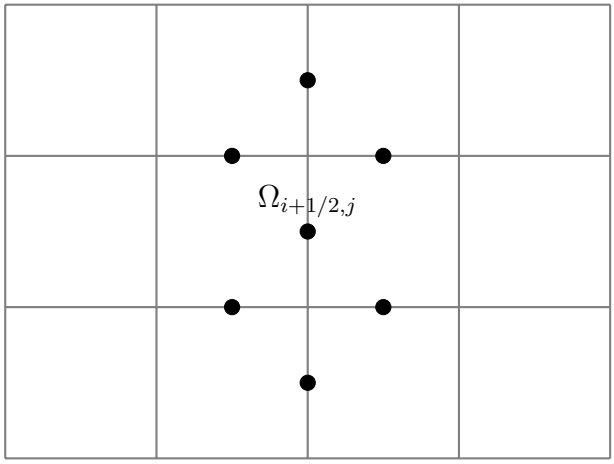


Figure 2.1: Nodes that necessary to compute potential $\Omega_{i+1/2,j}$

divergence value

$$(\nabla \cdot \mathbf{B})_{i+1/2,j+1/2} = \frac{(B_1)_{i+1,j} + (B_1)_{i+1,j+1} - (B_1)_{i,j} - (B_1)_{i,j+1}}{2\Delta x} + \frac{(B_2)_{i,j+1} + (B_2)_{i+1,j+1} - (B_2)_{i,j} - (B_2)_{i+1,j}}{2\Delta y} \tag{2.4.9}$$

vanishes if the initial magnetic field is divergence-free.

2.4.2 Divergence cleaning technique

In [40] the divergence constraint for the electric field $\mathbf{E}$ in the Maxwell equation has been coupled with the evolution equation for $\mathbf{E}$ by introducing a new unknown scalar function ψ. Different possibilities for this correction technique have been examined in [38] which classified techniques for an elliptic, a parabolic and a hyperbolic equation. The equations related to the magnetic field in Eq (2.3.1) are replaced by

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{B} - \mathbf{B} \otimes \mathbf{u}) + \nabla \psi = \mathbf{0}, \tag{2.4.10}$$

$$\mathcal{D}(\psi) + \nabla \cdot \mathbf{B} = 0, \tag{2.4.11}$$

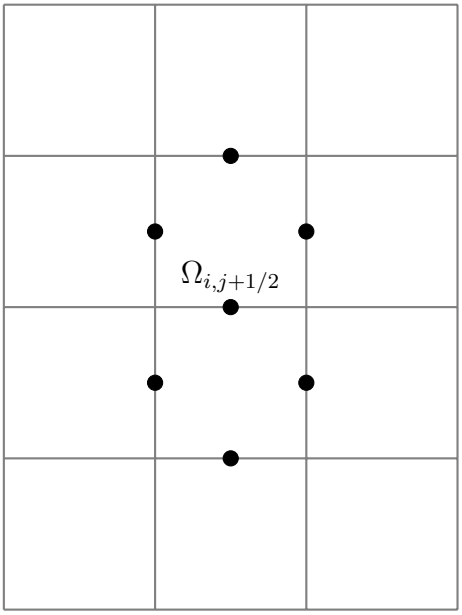


Figure 2.2: Nodes that necessary to compute potential $\Omega_{i,j+1/2}$

where $\mathcal{D}$ is a linear differential operator. Under the assumption of smooth solutions, we obtain

$$\frac{\partial(\nabla \cdot \mathbf{B})}{\partial t} + \Delta\psi = 0, \tag{2.4.12}$$

$$\frac{\partial \mathcal{D}(\nabla \cdot \mathbf{B})}{\partial t} + \Delta\mathcal{D}(\psi) = 0, \tag{2.4.13}$$

$$\frac{\partial \mathcal{D}(\psi)}{\partial t} + \frac{\partial(\nabla \cdot \mathbf{B})}{\partial t} = 0, \tag{2.4.14}$$

$$\Delta\mathcal{D}(\psi) + \Delta(\nabla \cdot \mathbf{B}) = 0. \tag{2.4.15}$$

Cancelling the term $\frac{\partial(\nabla \cdot \mathbf{B})}{\partial t}$ and $\Delta\mathcal{D}(\psi)$, we have

$$\frac{\partial \mathcal{D}(\nabla \cdot \mathbf{B})}{\partial t} - \Delta(\nabla \cdot \mathbf{B}) = 0, \tag{2.4.16}$$

$$\frac{\partial \mathcal{D}(\psi)}{\partial t} - \Delta\psi = 0. \tag{2.4.17}$$

In other words, $\nabla \cdot \mathbf{B}$ and ψ satisfy the same equation for any choice of $\mathcal{D}$.

Our main goal is to make $\mathcal{D}(\psi)$ close to zero so that $\nabla \cdot \mathbf{B}$ become zero by Eq (2.4.11). There are four possibilities of operator $\mathcal{D}$: *elliptic*, *parabolic*,

hyperbolic and *mixed type*.

If we define

$$\mathcal{D}(\psi) = 0, \tag{2.4.18}$$

then ψ is just a Lagrangian multiplier. We apply the following two step method in the operator splitting manner.

Step 1) Update all variables with (2.3.1) using the base scheme. Denote the resulting magnetic field as $\mathbf{B}^*$.

Step 2) Calculate the ψ^* by solving the Poisson equation

$$-\Delta\psi^* = \frac{1}{\Delta t}(\nabla \cdot \mathbf{B}^* - \nabla \cdot \mathbf{B}^n) = \frac{1}{\Delta t}\nabla \cdot \mathbf{B}^*. \qquad \text{(By Eq (2.4.12))}$$

Step 3) Compute $\mathbf{B}^{n+1}$ by carrying out

$$\mathbf{B}^{n+1} = \mathbf{B}^* - \Delta t \nabla \psi^*. \qquad \text{(By Eq (2.4.10))}$$

Therefore $\mathbf{B}^{n+1}$ is the projection of $\mathbf{B}^*$ onto the space of divergence-free fields, as introduced in [41]. This correction technique is called as *elliptic* correction, since the Poisson equation which is classified as an elliptic PDE is solved in the correction process. The method has been widely used in a field of incompressible Navier-Stokes equation and usually called *projection* method.

A *parabolic* correction is obtained by choosing

$$\mathcal{D}(\psi) = \frac{1}{c_p^2}\psi \quad \text{with} \quad c_p \in (0, \infty), \tag{2.4.19}$$

since Eq (2.4.17) becomes the heat equation

$$\frac{\partial \psi}{\partial t} - c_p^2 \Delta \psi = 0. \tag{2.4.20}$$

It is worth to note that no additional scalar field ψ is required in the parabolic case, since substituting the Eq (2.4.19) into the Eq (2.4.10) and the Eq

(2.4.11) leads to

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{B} - \mathbf{B} \otimes \mathbf{u}) = c_p^2 \nabla (\nabla \cdot \mathbf{B}). \tag{2.4.21}$$

The most interesting choice for the correction operator is the *hyperbolic* correction where

$$\mathcal{D}(\psi) = \frac{1}{c_h^2} \frac{\partial \psi}{\partial t} \quad \text{with} \quad c_h \in (0, \infty). \tag{2.4.22}$$

In this case, Eq (2.4.17) brings to the wave equation

$$\frac{\partial^2 \psi}{\partial t^2} - c_h^2 \Delta \psi = 0. \tag{2.4.23}$$

Hence a local divergence error are propagated to the boundary with the finite wave speed $c_h > 0$.

In order to achieve the *mixed* correction, we need to define

$$\mathcal{D}(\psi) = \frac{1}{c_h^2} \frac{\partial \psi}{\partial t} + \frac{1}{c_p^2} \psi. \tag{2.4.24}$$

Now Eq (2.4.17) is equal to the equation

$$\frac{\partial^2 \psi}{\partial t^2} + \frac{c_h^2}{c_p^2} \frac{\partial \psi}{\partial t} - c_h^2 \Delta \psi = 0 \tag{2.4.25}$$

and offers both diffusion and propagation of the divergence error. The divergence constraint Eq (2.4.11) takes the form

$$\frac{\partial \psi}{\partial t} + c_h^2 \nabla \cdot \mathbf{B} = -\frac{c_h^2}{c_p^2} \psi. \tag{2.4.26}$$

The whole process is as follows :

Step 1) Compute the numerical flux $\mathbf{F}$ using a base scheme.

Step 2) Modify the numerical flux $\mathbf{F}$ by adding the following term to

magnetic flux part :

$$\text{magnetic part of flux } \mathbf{F} \quad += \quad \nabla\psi. \tag{2.4.27}$$

A central difference is used to approximate $\nabla\psi$.

Step 3) Update the scalar field ψ by solving

$$\frac{\partial\psi}{\partial t} + c_h^2 \nabla \cdot \mathbf{B} = 0. \tag{2.4.28}$$

Denote the updated value as ψ^*.

Step 4) Update the scalar field ψ by exactly solving

$$\frac{\partial\psi}{\partial t} = -\frac{c_h^2}{c_p^2}\psi, \tag{2.4.29}$$

with the initial condition ψ^*. Hence,

$$\psi^{n+1} = \exp\left(-\Delta t \frac{c_h^2}{c_p^2}\right)\psi^*. \tag{2.4.30}$$

In our applications, we set the parameters in order to fix the amplification factor :

$$\exp\left(-\Delta t \frac{c_h^2}{c_p^2}\right) = 0.5.$$

Step 5) Update the physical vector $\mathbf{q}$ using the precomputed flux $\mathbf{F}$.

Chapter 3

Wavelet-based Adaptation Strategy with Finite Difference WENO scheme

3.1 Finite Difference WENO scheme

In this section, we review the FD-WENO(Finite Difference WENO) scheme for solving hyperbolic conservation laws. FD-WENO scheme is renowned for its high order accuracy, efficiency and robustness as well as its straightforward extension to multidimensional space via dimensional splitting.

3.1.1 Characteristic Decomposition

In this section, we describe the characteristic decomposition method for one-dimensional hyperbolic conservation laws :

$$\frac{\partial u}{\partial t}(x,t) + \frac{\partial f}{\partial x}(u(x,t)) = 0, \tag{3.1.1}$$

where u and $f(u)$ are vector-valued functions. It is enough to consider only one-dimensional situation since multi-dimensional problem can be solved in a dimensional splitting manner. The Equation (3.1.1) can be rewritten as a

quasi-linear form :

$$\frac{\partial u}{\partial t}(x,t) + \frac{\partial f}{\partial u}\frac{\partial u}{\partial x}(x,t) = 0, \tag{3.1.2}$$

with the Jacobian matrix $\dfrac{\partial f}{\partial u} = \left(\dfrac{\partial f_i}{\partial u_j}\right)$.

By the hyperbolicity of the system, the Jacobian matrix $\dfrac{\partial f}{\partial u}$ can be diagonalized as

$$\frac{\partial f}{\partial u} = R\Lambda L, \quad \Lambda = diag(\lambda^1, \lambda^2, \cdots, \lambda^n),$$

where n is the number of equations, Λ is the diagonal matrix whose entries are the eigenvalues of the Jacobian matrix, and R and L are the matrices of the right eigenvectors and the left eigenvectors, respectively.

We consider the computational domain $[a, b]$ with $N + 1$ uniform grid points as follows :

$$a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b.$$

The finite difference WENO scheme for Equation (3.1.1) can be written in the following semi-discrete flux-difference form :

$$\frac{du}{dt}(x_i) = -\frac{1}{\Delta x}(\widehat{f}_{i+1/2} - \widehat{f}_{i-1/2}).$$

To obtain the numerical flux $\widehat{f}_{i+1/2}$, the Algorithm 1 is performed.

Algorithm 1 Computation of $\widehat{f}_{i+1/2}$

1. Compute the physical flux at each grid point :

$$f_i = f(u_i).$$

2. Compute the average value $u_{i+1/2}$ by taking arithmetic average or Roe average:

$$u_{i+1/2} = \frac{1}{2}(u_i + u_{i+1}) \quad \text{or} \quad u_{i+1/2} = Roe(u_i, u_{i+1}).$$

3. Construct the right and left eigenvectors at $x_{i+1/2}$:

$$R_{i+1/2} = R(u_{i+1/2}) \quad \text{and} \quad L_{i+1/2} = L(u_{i+1/2}).$$

4. Project the physical quantities and fluxes into the characteristic fields :

$$U_j = L_{i+1/2} u_j \quad \text{and} \quad F_j = L_{i+1/2} f_j.$$

for all $j = i-r, \cdots, i+s$ necessary to compute a numerical approximation at $x_{i+1/2}$. In the case of the fifth order WENO interpolation, $r = 2, s = 3$.

5. Perform LF flux splitting for each component of the characteristic variables :

$$F_j^{\pm} = \frac{1}{2}(F_j \pm \alpha \odot U_j),$$

where $\alpha = (\max_i |\lambda_i^1|, \max_i |\lambda_i^2|, \cdots, \max_i |\lambda_i^n|)$ and $\odot$ is a component-wise product.

6. Interpolate the interfacial values $F_{i+1/2}^+$ and $F_{i+1/2}^-$ by using F_j^+ and F_j^- :

$$F_{i+1/2}^+ = \mathcal{I}^{left}(F_{j-r}^+, \cdots, F_{j+s}^+),$$
$$F_{i+1/2}^- = \mathcal{I}^{right}(F_{j-r}^-, \cdots, F_{j+s}^-).$$

7. Project back into the physical domain :

$$\widehat{f}_{i+1/2} = R_{i+1/2}(F_{i+1/2}^+ + F_{i+1/2}^-).$$

3.1.2 WENO-type Approximations

To achieve an accurate and robust characteristic decomposition scheme, an interpolation(reconstruction) procedure that has high order accuracy in smooth region and guarantees stable results when shock waves appear needs to be devised. WENO reconstruction is the one of popular schemes that satisfy the above properties [3].

The main idea of the WENO interpolation is to design non-linear weights, w_r, according to the smoothness of each sub-stencil. Smoothness is basically measured by *smoothness indicator* involving L^2 norms for all of the derivatives of the interpolation polynomials. In order to capture both continuous and discontinuous profile elaborately, it is crucial to design sophisticated smoothness indicators so that they monitor the region where shocks occur and recover the ideal weight, d_r, near smooth extrema. Various variations of the WENO-JS [3] interpolation method are shown in WENO-M [6], WENO-Z [7, 42], WENO-NS [8], according to the smoothness indicators. In our implementation, WENO-JS is chosen with $\epsilon = 10^{-6}$ to prevent from dividing by zero.

After the weights denoted w_r are determined, the value at the cell interface is computed by the weighted sum of each sub-stencil interpolation polynomial. The WENO interpolation procedure is described in Algorithm 2. It is trivial that $u^{left}_{i+1/2} = \mathcal{I}^{right}_{WENO}(u_{i+3}, u_{i+2}, u_{i+1}, u_i, u_{i-1}, u_{i-2})$ by the symmetric property of the WENO interpolation.

Algorithm 2 Fifth order WENO interpolation, $\mathcal{I}^{right}_{WENO}$, at $x_{i+1/2}$.

Input : $u_{i-2}, u_{i-1}, u_i, u_{i+1}, u_{i+2}, u_{i+3}$
Procedure :

1. Compute smoothness indicator $\alpha_r, \quad r = 0, 1, 2$:

$$\alpha_r = \frac{d_r}{(\beta_r + \epsilon)^2},$$

where

$$\begin{aligned}
\beta_0 &= \frac{13}{12}(u_{i+1} - 2u_{i+2} + u_{i+3})^2 + \frac{1}{4}(3u_{i+1} - 4u_{i+2} + u_{i+3})^2, \\
\beta_1 &= \frac{13}{12}(u_i - 2u_{i+1} + u_{i+2})^2 + \frac{1}{4}(u_i - u_{i+2})^2, \\
\beta_2 &= \frac{13}{12}(u_{i-1} - 2u_i + u_{i+1})^2 + \frac{1}{4}(u_{i-1} - 4u_i + 3u_{i+1}),
\end{aligned}$$

the ideal weights are given by

$$d_0 = 0.3, \; d_1 = 0.6, \; d_2 = 0.1.$$

2. Determine the weights according to the each sub-stencil by normalizing α_r :

$$w_r = \frac{\alpha_r}{\alpha_0 + \alpha_1 + \alpha_2}, \quad r = 0, 1, 2.$$

3. Evaluate the interfacial value :

$$u^{right}_{i+1/2} = \sum_{r=0}^{2} \sum_{k=0}^{2} w_r c_{r-1,k} u_{i+1+k-r},$$

where the coefficients are given by

$$\begin{array}{lll}
c_{-1,0} = 11/6, & c_{-1,1} = -7/6, & c_{-1,2} = 1/3, \\
c_{0,0} = 1/3, & c_{0,1} = 5/6, & c_{0,2} = -1/6, \\
c_{1,0} = -1/6, & c_{1,1} = 5/6, & c_{1,2} = 1/3.
\end{array}$$

Output : $u^{right}_{i+1/2} = \mathcal{I}^{right}_{WENO}(u_{i-2}, u_{i-1}, u_i, u_{i+1}, u_{i+2}, u_{i+3})$

3.2 Wavelet Analysis

3.2.1 Multi-resolution Approximations

Our purpose is to decompose functions in $L^2(\mathbb{R})$ according to different resolution levels. We start with the mathematical definition of *Multi-Resolution*

Approximation introduced by [43], [44] and [45]. We first need to define *Riesz basis*.

Definition 3.2.1 (Riesz basis)**.** Assume that V is a subspace of $L^2(\mathbb{R})$. We call a set of functions

$$\{e_n \in V \mid n \in \mathbb{Z}\},$$

a *Riesz basis* of V if there exist $A > 0$ and $B > 0$ such that any function $f \in V$ can be uniquely decomposed into

$$f(x) = \sum_{n=-\infty}^{\infty} a_n e_n(x),$$

where $a_n \in \mathbb{R}$ and satisfies the following inequality

$$A \|f\|_{L^2}^2 \leq \sum_{n=-\infty}^{\infty} |a_n|^2 \leq B \|f\|_{L^2}^2 .$$

Now, we are ready to define the *MRA*.

Definition 3.2.2 (MRA)**.** A sequence $\{V_j\}_{j\in\mathbb{Z}}$ of closed subspace of $L^2(\mathbb{R})$ is a *multi-resolution approximation* if the following 6 properties are satisfied :

$$f\left(x - \frac{k}{2^j}\right) \in V_j \quad \text{for all} \quad j,k \in \mathbb{Z},\ f(x) \in V_j,$$

$$V_j \subset V_{j+1} \quad \text{for all} \quad j \in \mathbb{Z},$$

$$f(2x) \in V_{j+1} \quad \text{for all} \quad j \in \mathbb{Z},\ f(x) \in V_j,$$

$$\lim_{j\to-\infty} V_j = \bigcap_{-\infty}^{\infty} V_j = \{0\},$$

$$\lim_{j\to\infty} V_j = \overline{(\bigcup_{j=-\infty}^{\infty} V_j)} = L^2(\mathbb{R}),$$

and there exists $\phi(x) \in L^2(\mathbb{R})$ such that $\{\phi(x-k)\}_{k\in\mathbb{Z}}$ is a *Riesz basis* of V_0.

Remark 3.2.3. 1. The subindex j is the *resolution level*. The functions

in V_j have more detailed information as the resolution level j increases.

2. For any $f \in L^2(\mathbb{R})$, we denote its *orthogonal projection* into the subspace V_j of $L^2(\mathbb{R})$ by $P_{V_j} f$. It is trivial that

$$\lim_{j\to-\infty} \left\| P_{V_j} f \right\|_{L^2} = 0 \quad \text{and} \quad \lim_{j\to\infty} \left\| f - P_{V_j} \right\|_{L^2} = 0. \tag{3.2.1}$$

Roughly speaking, nothing is left when the resolution level is significantly small, and everything can be recovered when the resolution is sufficiently large.

3. It is easy to check that :

For any $j \in \mathbb{Z}$, the set $\{\phi_{j,k}\}_{k\in\mathbb{Z}}$ forms a Riesz basis of V_j,

where

$$\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k).$$

Since $V_0 \subset V_1$ and $\phi \in V_0$, we can represent ϕ as a linear combination of the elements of $\{\phi_{1,k}\}$:

$$\begin{aligned} \phi(x) &= \sum_{k=-\infty}^{\infty} h_k \phi_{1,k}(x) \\ &= \sqrt{2} \sum_{k=-\infty}^{\infty} h_k \phi(2x - k), \end{aligned} \tag{3.2.2}$$

where $h_k \in \mathbb{R}$. We call ϕ *scaling function* and the Eq (3.2.2) *scaling equation*. The sequence of coefficients $\{h_k\}_{k\in\mathbb{Z}}$ is called the *filer of scaling function* ϕ. By applying Fourier transformation to Eq (3.2.2), we obtain

$$\widehat{\phi}(\omega) = \widehat{\phi}\left(\frac{\omega}{2}\right) \frac{1}{\sqrt{2}} \sum_{k=-\infty}^{\infty} h_k e^{-iwk/2}. \tag{3.2.3}$$

By introducing, hence, the *symbol* $\mathbf{h}$ of the filter h by

$$\mathbf{h}(\omega) = \frac{1}{\sqrt{2}} \sum_{n\in\mathbb{Z}} h_n e^{-i\omega n}, \tag{3.2.4}$$

we have the another version of the *scaling equation* :

$$\widehat{\phi}(\omega) = \widehat{\phi}\left(\frac{\omega}{2}\right) \mathbf{h}\left(\frac{\omega}{2}\right). \tag{3.2.5}$$

The symbol $\mathbf{h}$ represents some properties related to the corresponding scaling function ϕ.

Theorem 3.2.4. *A family of functions $\{\phi(x-n)\}_{n\in\mathbb{Z}}$ is a Riesz basis of the space V_0 if and only if there exist $A > 0$ and $B > 0$ such that*

$$\frac{1}{B} \leq \sum_{k\in\mathbb{Z}} \left|\widehat{\phi}(\omega + 2k\pi)\right|^2 \leq \frac{1}{A}, \quad \textit{for all } \omega \in [-\pi, \pi],$$

where $\widehat{\phi}$ is the Fourier transform of ϕ.

Proof. See S.Mallat [45]. □

Let f be a function in $L^2(\mathbb{R})$ and $\{V_j\}_{j\in\mathbb{Z}}$ be an MRA. In every subspace V_j for every $j \in \mathbb{Z}$, in order to approximate f as its orthogonal projection $P_{V_j} f$, we need an orthonormal basis of V_j.

Definition 3.2.5 (Orthogonal scaling function)**.** We call a scaling function ϕ of an MRA $\{V_j\}_{j\in\mathbb{Z}}$ *orthogonal(orthonormal)* if $\{\phi_{j,n}\}_{n\in\mathbb{Z}}$ is an orthogonal(orthonormal) basis of V_j for all $j \in \mathbb{Z}$.

Lemma 3.2.6. *Let $\{V_j\}_{j\in\mathbb{Z}}$ be an MRA and ϕ be a scaling function. Then ϕ is orthonormal if and only if*

$$\sum_{k=-\infty}^{\infty} \left|\widehat{\phi}(\omega + 2k\pi)\right|^2 = 1, \quad \textit{for } \omega \in \mathbb{R}. \tag{3.2.6}$$

3.2.2 Orthogonal Wavelets

Let f be an $L^2(\mathbb{R})$ function and $\{V_j\}_{j\in\mathbb{Z}}$ be an MRA. Suppose that W_j is an orthogonal complement of V_j in V_{j+1} :

$$V_{j+1} = V_j \ \oplus \ W_j. \tag{3.2.7}$$

The orthogonal projection of f on V_{j+1} can be decomposed as the sum of the orthogonal projection on V_j and W_j:

$$P_{V_{j+1}} f = P_{V_j} f + P_{W_j} f.$$

Here the complement $P_{W_j} f$ contains the detailed information of f which can be described at the resolution level $j+1$ but cannot be expressed at the resolution level j. By performing the decomposition (3.2.7) repeatedly, we have

$$V_l = W_{l-1} \oplus W_{l-2} \oplus \cdots \oplus W_{j+1} \oplus W_j \oplus V_j \quad \text{for any } j < l.$$

Because $\{V_j\}_{j\in\mathbb{Z}}$ is an MRA, by letting $j \to -\infty$ and $l \to \infty$, we obtain

$$L^2(\mathbb{R}) = \bigoplus_{j=-\infty}^{\infty} W_j.$$

In other words, an arbitrary function $f \in L^2(\mathbb{R})$ can be represented as the superposition of details of all resolution levels,

$$f = \sum_{j=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \langle f, \psi_{j,n} \rangle \psi_{j,n}.$$

We call the function ψ a *wavelet* and W_j a *wavelet space* of the resolution level j.

Definition 3.2.7 (Orthonormal wavelet)**.** A wavelet ψ is orthonormal if $\{\psi_{j,k}\}_{j,k\in\mathbb{Z}}$ is an orthonormal basis of $L^2(\mathbb{R})$.

Since $\psi \in W_0 \subset V_1$, we can represent ψ with the linear combination of

the basis elements $\{\phi_{1,k}\}_{k\in\mathbb{Z}}$,

$$\psi(x) = \sqrt{2}\sum_{n=-\infty}^{\infty} g_n \phi(2x-n).$$

We call the sequence g_n the *filter of the wavelet of* ψ. Also, the *symbol* $\mathbf{g}$ denotes the Fourier series of the filter g_n :

$$\mathbf{g}(\omega) = \frac{1}{\sqrt{2}}\sum_{n=-\infty}^{\infty} g_n \exp(-in\omega).$$

As one can expect, the properties of the wavelet ψ depend on its filter g_n and the scaling function ϕ.

3.2.3 Constructing Wavelets

When we want to construct a wavelet ψ, we mainly consider the number of vanishing moments of ψ and its size of support.

Definition 3.2.8 (Vanishing moments)**.** A function ψ has p *vanishing moments* if

$$\int_{-\infty}^{\infty} t^k \psi(t)dt = 0, \quad \text{for } 0 \le k < p.$$

In other words, this means that ψ is orthogonal to any polynomial of degree less than p. If f is locally C^k, then it is well approximated by a Taylor polynomial of degree k over a small interval. If $k < p$, then wavelets are orthogonal to this Taylor polynomial, and thus produce small amplitude coefficients at fine resolution levels [45].

In many applications, the number of non-zero coefficients of the filter of a scaling function ϕ directly affects the computational cost. From the following lemma, we can know a connection between the number of non-zero coefficients of the filter and the support size of the scaling function.

Lemma 3.2.9 (Compact support)**.** *The scaling function ϕ has a compact support if and only if the filter h has a compact support. Furthermore, their*

supports are equal.

Proof. See S.Mallat [45]. □

3.2.4 Biorthogonal Wavelets

The construction of compactly supported orthogonal wavelet with certain regularity is totally dependent on the design of the symbol **h**. This is quite a burden to **h**. By replacing the orthogonality by the biorthogonality with two dual functions $\widetilde{\phi}, \widetilde{\psi}$, we may relieve the burden on a single **h**. These dual functions are called a *dual scaling function* and a *dual wavelet function* respectively, whose corresponding symbols are $\widetilde{\mathbf{h}}$ and $\widetilde{\mathbf{g}}$. We have the following four refinement relations :

$$\begin{aligned}
\phi(x) &= \sqrt{2}\sum_{n=-\infty}^{\infty} h_n \phi(2x-n), \quad \widetilde{\phi}(x) = \sqrt{2}\sum_{n=-\infty}^{\infty} \widetilde{h}_n \widetilde{\phi}(2x-n), \\
\psi(x) &= \sqrt{2}\sum_{n=-\infty}^{\infty} g_n \phi(2x-n), \quad \widetilde{\psi}(x) = \sqrt{2}\sum_{n=-\infty}^{\infty} \widetilde{g}_n \widetilde{\phi}(2x-n).
\end{aligned}$$

The biorthogonality requires that

$$\begin{aligned}
\left\langle \phi(x),\, \widetilde{\phi}(x-n) \right\rangle = \left\langle \psi(x),\, \widetilde{\psi}(x-n) \right\rangle = \delta_{n,0}, \\
\left\langle \phi(x),\, \widetilde{\psi}(x-n) \right\rangle = \left\langle \psi(x),\, \widetilde{\phi}(x-n) \right\rangle = 0.
\end{aligned} \tag{3.2.8}$$

Definition 3.2.10 (Biorthogonality)**.** We call a set of scaling functions and wavelets, $\{\phi, \psi, \widetilde{\phi}, \widetilde{\psi}\}$, a family of *biorthogonal scaling functions* and *wavelets* if it satisfies Eq (3.2.8).

Through a simple algebra, we obtain the relations among the filters $h_n, g_n, \widetilde{h}_n$ and $\widetilde{g}_n$:

$$\begin{aligned}
\sum_{k=-\infty}^{\infty} h_k \widetilde{h}_{k-2n} = \sum_{k=-\infty}^{\infty} g_k \widetilde{g}_{k-2n} = \delta_{n,0}, \\
\sum_{k=-\infty}^{\infty} h_k \widetilde{g}_{k-2n} = \sum_{k=-\infty}^{\infty} g_k \widetilde{h}_{k-2n} = 0, \quad \text{for all } n \in \mathbb{Z}.
\end{aligned} \tag{3.2.9}$$

By taking a discrete Fourier transformation to the first relation in Eq (3.2.9), we have

$$\sum_{n=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} h_k\widetilde{h}_{k-2n}e^{-iwn} = \sum_{n=-\infty}^{\infty}\delta_{n,0}e^{-iwn} = 1.$$

By interchanging the summations and computing simple algebras, we obtain

$$\begin{aligned}
\sum_{k=-\infty}^{\infty} h_k \sum_{n=-\infty}^{\infty}\widetilde{h}_{k-2n}e^{-iwn} &= \frac{1}{2}\sum_{k=-\infty}^{\infty} h_k \sum_{n=-\infty}^{\infty}\left(\widetilde{h}_n + (-1)^{n+k}\widetilde{h}_n\right)e^{iw(n-k)/2} \\
&= \frac{1}{2}\sum_{k=-\infty}^{\infty} h_k \sum_{n=-\infty}^{\infty}\left(\widetilde{h}_n e^{iwn/2}e^{-iwk/2} + (-1)^k\widetilde{h}_n e^{i(w+2\pi)n/2}e^{-iwk/2}\right) \\
&= \frac{1}{\sqrt{2}}\sum_{k=-\infty}^{\infty} h_k\left(\overline{\widetilde{\mathbf{h}}\left(\frac{\omega}{2}\right)}e^{-iwk/2} + (-1)^k\overline{\widetilde{\mathbf{h}}\left(\frac{\omega}{2}+\pi\right)}e^{-iwk/2}\right) \\
&= \mathbf{h}\left(\frac{\omega}{2}\right)\overline{\widetilde{\mathbf{h}}\left(\frac{\omega}{2}\right)} + \mathbf{h}\left(\frac{\omega}{2}+\pi\right)\overline{\widetilde{\mathbf{h}}\left(\frac{\omega}{2}+\pi\right)}.
\end{aligned}$$

Hence we have the following equations of symbols

$$\begin{aligned}
&\overline{\mathbf{h}(\omega)}\widetilde{\mathbf{h}}(\omega) + \overline{\mathbf{h}(\omega+\pi)}\widetilde{\mathbf{h}}(\omega+\pi) = 1, \quad \overline{\mathbf{g}(\omega)}\widetilde{\mathbf{g}}(\omega) + \overline{\mathbf{g}(\omega+\pi)}\widetilde{\mathbf{g}}(\omega+\pi) = 1, \\
&\overline{\mathbf{h}(\omega)}\widetilde{\mathbf{g}}(\omega) + \overline{\mathbf{h}(\omega+\pi)}\widetilde{\mathbf{g}}(\omega+\pi) = 0, \quad \overline{\mathbf{g}(\omega)}\widetilde{\mathbf{h}}(\omega) + \overline{\mathbf{g}(\omega+\pi)}\widetilde{\mathbf{h}}(\omega+\pi) = 0,
\end{aligned} \tag{3.2.10}$$

for all $\omega \in \mathbb{R}$.

Definition 3.2.11. If a group of filter $(h_n, g_n, \widetilde{h}_n, \widetilde{g}_n)$ satisfies Eq (3.2.9), then we call it a family of *biorthogonal filters*. Moreover, if a group of symbols $(\mathbf{h}, \mathbf{g}, \widetilde{\mathbf{h}}, \widetilde{\mathbf{g}})$ satisfies Eq (3.2.10), then we call it a family of *biorthogonal symbols*.

Let f be a function in $L^2(\mathbb{R})$ and $\{V_j\}_{j\in\mathbb{Z}}$ be the MRA generated by ϕ and $\{W_j\}_{j\in\mathbb{Z}}$ be wavelet spaces generated by ψ. We have the corresponding dual MRA $\{\widetilde{V}_j\}_{j\in\mathbb{Z}}$ and dual wavelet spaces $\{\widetilde{W}_j\}_{j\in\mathbb{Z}}$ generated by $\widetilde{\phi}$ and $\widetilde{\psi}$, respectively. In a biorthogonal case, W_j is not orthogonal to each other for

different $j \in \mathbb{Z}$. Instead, we have

$$V_j \perp \widetilde{W_j} \quad \text{and} \quad W_j \perp \widetilde{V_j}.$$

Therefore, f can be expressed with the basis $\{\psi_{j,k}\}_{j,k\in\mathbb{Z}}$

$$f = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \left\langle f, \widetilde{\psi}_{j,k} \right\rangle \psi_{j,k},$$

which is not an orthogonal decomposition.

3.2.5 Interpolating Scaling Function

We will construct *ISF(Interpolating scaling function)* on the real line [46] [47]. Let N be a positive integer. We interpolate the Kronecker delta sequence $\{\delta_{n,0}\}_{n\in\mathbb{Z}}$ at the integers to a function on the binary rationals by repeating the following process. We call this the *iterative interpolating process*. Since

Algorithm 3 Construction of ISF, DD_N

for $j = 0, 1, 2, \cdots$ **do**

Interpolate the vales at points $\dfrac{k+1/2}{2^j}$ for all $k \in Z$ with the Lagrangian interpolation using symmetric $2N$ points.

end for

this function is uniformly continuous on the whole binary rationals, it can be uniquely extended to a function ϕ on $\mathbb{R}$. We denote the function ϕ as DD_N.

The construction with the iterative interpolating process tells us that DD_N is even symmetric :

$$DD_N(x) = DD_N(-x), \quad t \in \mathbb{R}.$$

Theorem 3.2.12. *If $p(x)$ is a polynomial of degree less than $2N$, then $p(x)$*

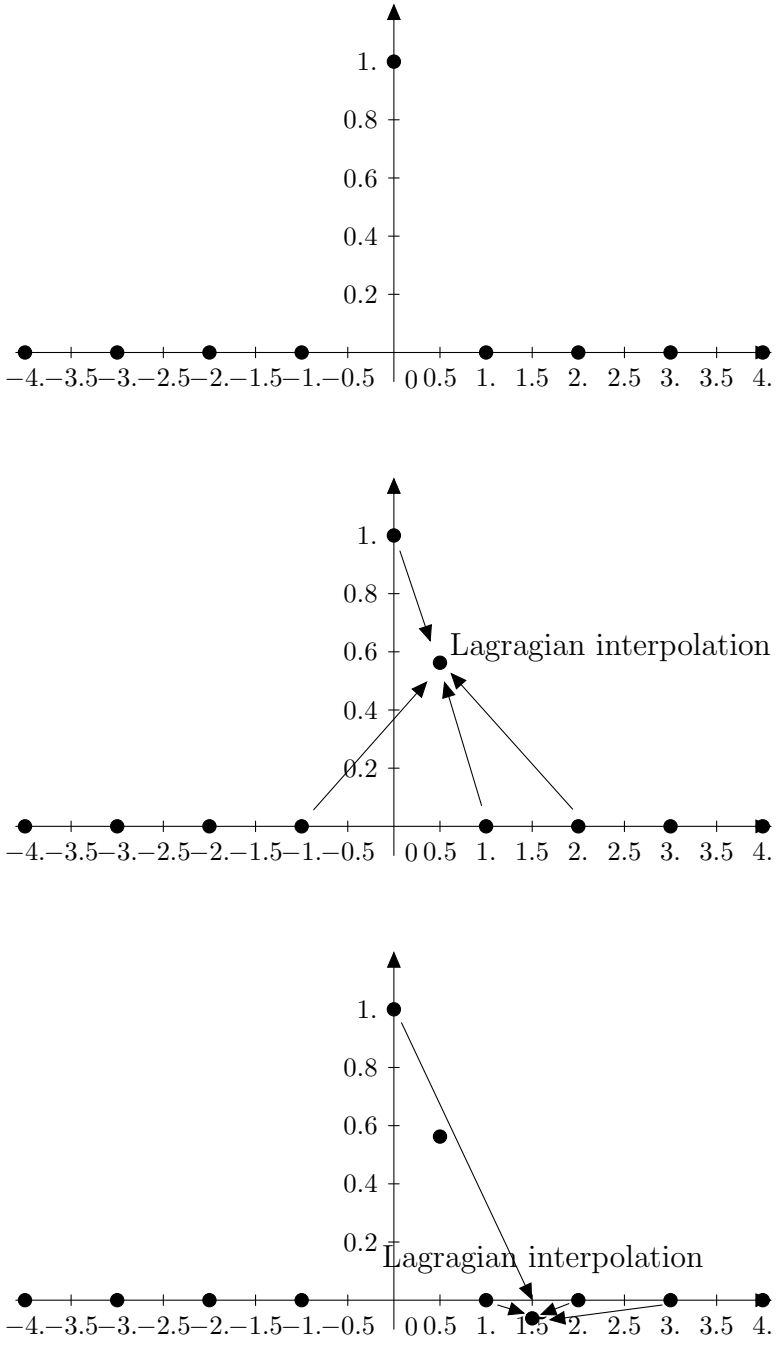


Figure 3.1: Description of an iterative interpolating process with $N = 2$.

can be reproduced by the $\{DD_N(t-n)\}_{n\in\mathbb{Z}}$,

$$p(x) = \sum_{n=-\infty}^{\infty} p(n) DD_N(x-n).$$

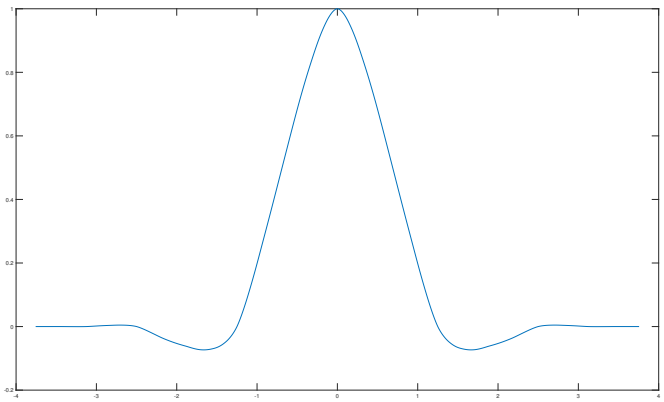


Figure 3.2: Profile of DD_2.

As one may expect that the ISF DD_N has a compact support, the following theorem gives us the interval of support of the function.

Theorem 3.2.13 (Support of DD_N)**.** *For any $N \in \mathbb{N}$, DD_N vanishes outside of $(-2N+1, 2N-1)$.*

Proof. Consider a sequence $\{x_n\}_{n\geq 0}$ satisfying the recurrence

$$x_{n+1} = x_n + \frac{2N-1}{2^{n+1}}, \quad x_0 = 0.$$

For a given $n \in \mathbb{N}$, we can notice that $DD_N(x)$ vanishes outside of $[-x_n, x_n]$ at the resolution level n. By simple computations, we obtain the general expression of the sequence

$$x_n = 2N - 1 - \frac{2N-1}{2^n}.$$

By letting $n \to \infty$, we conclude that DD_N vanishes outside of $(-2N+1, 2N-1)$. □

Furthermore, it is well known that DD_N is continuously differentiable

[47]. The scaling equation of DD_N is given as

$$DD_N(x) = \sum_{k=-\infty}^{\infty} h_k DD_N(2x-k).$$

By the interpolating property of DD_N, the filter h_k yields

$$h_k = DD_N\left(\frac{k}{2}\right).$$

Therefore, we have

$$DD_N\left(\frac{x}{2}\right) = \sum_{j=-\infty}^{\infty} DD_N\left(\frac{j}{2}\right) DD_N(x-j).$$

From the Theorem 3.2.13, since the size of support of DD_N is known, the infinite summation is reduced to the finite summation :

$$DD_N\left(\frac{x}{2}\right) = \sum_{j=-4N+3}^{4N-3} DD_N\left(\frac{j}{2}\right) DD_N(x-j). \tag{3.2.11}$$

By the definition of DD_N, we can easily compute the filter values

$$DD_N\left(\frac{2j+1}{2}\right) = (-1)^{N-j} \frac{\Pi_{k=0}^{2N-1}(k-N+1/2)}{(j+1/2)(N-j-1)!(N+j)!}, \quad \text{for } j \geq 2.$$

For examples, a few values of DD_2 can be displayed :

$$DD_2\left(\pm\frac{1}{2}\right) = \frac{9}{16},\ DD_2\left(\pm\frac{3}{2}\right) = -\frac{1}{16} \text{ and } DD_2\left(\pm\frac{2j+1}{2}\right) = 0, \quad \text{for } j \geq 2.$$

We assume that f is the extension of a discrete data set $\{f_i\}_{i\in\mathbb{Z}}$ by the iterative interpolation process with DD_2. Then we can write $f(x)$ as

$$f(x) = \sum_{j=-\infty}^{\infty} f_j DD_2(x-j). \tag{3.2.12}$$

Differentiating Eq (3.2.12), we achieve

$$f'(x) = \sum_{j=-\infty}^{\infty} f_j DD_2'(x-j).$$

Substituting x for 0, we obtain

$$f'(0) = -DD_2'(1)\left[f(1) - f(-1)\right] - DD_2'(2)\left[f(2) - f(-2)\right]. \tag{3.2.13}$$

Therefore, we need to compute $DD_2'(1)$ and $DD_2'(2)$ in order to approximate the derivative of f at the center position. The following lemma gives the $DD_2'(1)$ and $DD_2'(2)$.

Lemma 3.2.14.

$$DD_2'(1) = -\frac{2}{3} \quad \textit{and} \quad DD_2'(2) = \frac{1}{12}.$$

Proof. Define a sequence p_n approximating $f'(0)$ with the aid of a central difference :

$$p_n = \frac{f(2^{-n}) - f(-2^{-n})}{2 \cdot 2^{-n}}.$$

The iterative interpolation process tells us that

$$\begin{aligned} f(2^{-n}) &= -\frac{1}{16}f(-2\cdot 2^{-n}) + \frac{9}{16}f(0) + \frac{9}{16}f(2\cdot 2^{-n}) - \frac{1}{16}f(4\cdot 2^{-n}), \\ f(-2^{-n}) &= -\frac{1}{16}f(-4\cdot 2^{-n}) + \frac{9}{16}f(-2\cdot 2^{-n}) + \frac{9}{16}f(0) - \frac{1}{16}f(2\cdot 2^{-n}). \end{aligned}$$

For $n \geq 1$, hence

$$\begin{aligned} p_n &= \frac{f(-4\cdot 2^{-n}) - 10f(-2\cdot 2^{-n}) + 10f(2\cdot 2^{-n}) - f(4\cdot 2^{-n})}{32\cdot 2^{-n}} \\ &= \frac{5}{4}\frac{f(2\cdot 2^{-n}) - f(-2\cdot 2^{-n})}{4\cdot 2^{-n}} - \frac{1}{4}\frac{f(4\cdot 2^{-n}) - f(-4\cdot 2^{-n})}{8\cdot 2^{-n}} \\ &= \frac{5}{4}p_{n-1} - \frac{1}{4}p_{n-2}. \end{aligned} \tag{3.2.14}$$

The general solution of the difference equation Eq (3.2.14) is

$$p_n = c_1 + c_2 \left(\frac{1}{4}\right)^n . \tag{3.2.15}$$

In order to determine the coefficient c_1 and c_2, we use the initial conditions

$$p_{-1} = c_1 + 4c_2 = \frac{f(2) - f(-2)}{4} \quad \text{and} \quad p_0 = c_1 + c_2 = \frac{f(1) - f(-1)}{2}.$$

Rearranging this equations, we have

$$\begin{aligned} c_1 &= \frac{2}{3}\left(f(1) - f(-1)\right) - \frac{1}{12}\left(f(2) - f(-2)\right), \\ c_2 &= -\frac{1}{16}\left(f(1) - f(-1)\right) + \frac{1}{12}\left(f(2) - f(-2)\right). \end{aligned} \tag{3.2.16}$$

If we substitute Eq (3.2.16) into Eq (3.2.15) and let $n \to \infty$, we obtain

$$f'(0) = \frac{2}{3}\left(f(1) - f(-1)\right) - \frac{1}{12}\left(f(2) - f(-2)\right).$$

Therefore, we conclude that

$$DD_2'(1) = -\frac{2}{3} \quad \text{and} \quad DD_2'(2) = \frac{1}{12}.$$

□

G. Deslauriers and S. Dubuc [46] have provided a beautiful technique on computations of these derivatives for $N = 3$, which may be generalized for the higher N. The results are given as the Table (3.1).

3.3 Adaptive wavelet Collocation Method

In the ISF method, which was discussed in the previous section, we used a uniform grid to discretize field values. Only scaling functions of one resolution level are involved in the ISFM. In this section we prepare to introduce AWCM(Adaptive Wavelet Collocation Method) [9] [10] [48], considering not

i	$DD'_2(i)$	$DD'_3(i)$	$DD'_4(i)$
1	2/3	272/365	39296/49553
2	-1/12	-53/365	-76113/396424
3		16/1095	1664/49553
4		1/2920	- 2645/1189272
5			-128/743295
6			1/1189272

Table 3.1: Derivative filters $DD'_N(i)$

only scaling functions of one resolution level but also wavelets of different resolution levels.

Let $\{\phi, \psi, \widetilde{\phi}, \widetilde{\psi}\}$ be a family of biorthogonal scaling functions and wavelets. And let $\{V_j\}_{j\in\mathbb{Z}}$ and $\{W_j\}_{j\in\mathbb{Z}}$ be the corresponding MRA and the wavelet spaces, respectively. In general, W_j is not the orthogonal complement of V_j in V_{j+1}. Let $j_0, j_1 \in \mathbb{N}$ such that $j_0 < j_1$. We know the following relation

$$V_{j_1} = V_{j_0} \oplus W_{j_0} \oplus W_{j_0+1} \oplus \cdots \oplus W_{j_1-1}. \tag{3.3.1}$$

Note that operator $\oplus$ denotes not an orthogonal decomposition but a direct sum.

Let f be a function in $L^2(\mathbb{R})$. We use notations P_j and Q_j to express projections of f into the subspaces V_j and W_j of $L^2(\mathbb{R})$, respectively. Then from Eq (3.3.1), we have

$$P_{j_1} f = P_{j_0} f + \sum_{j=j_0}^{j_1-1} Q_j f. \tag{3.3.2}$$

In terms of the basis functions $\{\phi_{j_0,k}\}_{k\in\mathbb{Z}}$ and $\{\psi_{j,m}\}_{j,m\in\mathbb{Z}}$, Eq (3.3.2) can be rewritten as follows :

$$P_{j_1} f = \sum_{k\in\mathbb{Z}} \alpha_{j_0,k}\phi_{j_0,k} + \sum_{j=j_0}^{j_1-1} \sum_{m\in\mathbb{Z}} \beta_{j,m}\psi_{j,m},$$

where

$$\alpha_{j_0,k} = \left\langle f, \widetilde{\phi}_{j_0,k} \right\rangle \quad \text{and} \quad \beta_{j,m} = \left\langle f, \widetilde{\psi}_{j,m} \right\rangle.$$

We call $\alpha_{j_0,k}$ *scaling coefficient* and $\beta_{j,m}$ *wavelet coefficient*. We may think that scaling coefficients $\alpha_{j_0,k}$ represent *rough information* of $P_{j_1} f$ at the coarsest resolution level j_0 and the wavelet coefficients $\beta_{j,m}$ express the *detailed information* of $P_{j_1} f$ using various resolution levels from j_0 to $j_1 - 1$.

Our interest is focused on the wavelet coefficients. In the several applications involving numerical solutions to PDE or image processing, we may approximate a function or signal by discarding some terms in its wavelet decomposition whose wavelet coefficients are negligible. Wavelet coefficients are obtained by carrying out the fast wavelet transformations. In the AWCM, the *lifted interpolating wavelets* constructed by *lifting scheme* [49] [50] are selected as the wavelet basis function. We call this lifted interpolating wavelets *Donoho wavelets*.

3.3.1 Interpolating Wavelets

Let us first study about the interpolating wavelets introduced by Donoho [51]. For $N \in \mathbb{N}$, we define a function Do_N by

$$Do_N(x) = 2DD_N(2x - 1).$$

Assume $j \geq 0$ and define $W_j = \overline{\text{span}\,\{(Do_N)_{j,m} \,|\, m \in \mathbb{Z}\}}$. Note that

$$(Do_N)_{j,m}(x) = 2^{j/2} Do_N(2^j x - m) = \sqrt{2}(DD_N)_{j+1,2m+1}(x).$$

Hence, we know that

$$W_j \subset V_{j+1} = \overline{\text{span}\{(DD_N)_{j+1,k} \,|\, k \in \mathbb{Z}\}}.$$

Suppose we have two sequences $\{\alpha_{j,k}\}_{k\in\mathbb{Z}}$ and $\{\beta_{j,m}\}_{m\in\mathbb{Z}}$. Let f be a function written by

$$f = \sum_{k\in\mathbb{Z}} \alpha_{j,k}(DD_N)_{j,k} + \sum_{m\in\mathbb{Z}} \beta_{j,m}(Do_N)_{j,m}. \tag{3.3.3}$$

Since $V_j, W_j \subset V_{j+1}$, f lies in V_{j+1}. Therefore, we can represent f by

$$f = \sum_{k\in\mathbb{Z}} \alpha_{j+1,k}(DD_N)_{j+1,k}. \tag{3.3.4}$$

For a given integer j, we define a set of *dyadic rationals* of degree j,

$$\mathcal{K}_j = \left\{ \frac{k}{2^j} \middle| k \in \mathbb{Z} \right\}.$$

Then we have the nested relation,

$$\mathcal{K}_j \subset \mathcal{K}_{j+1}.$$

We define a set of complementary grid points $\mathcal{M}_j = \mathcal{K}_{j+1} - \mathcal{K}_j$. The grid set $\mathcal{K}_{j+1}$ of resolution level $j+1$ is composed of two disjoint sets, $\mathcal{K}_j$ and $\mathcal{M}_j$. We call $\mathcal{K}_j$ a *coarse part* and $\mathcal{M}_j$ a *fine part* in the resolution level $j+1$,

$$\mathcal{K}_{j+1} = \mathcal{K}_j \,\dot\cup\, \mathcal{M}_j.$$

Now we consider the relation between the coefficients in Eq (3.3.3) and Eq (3.3.4). By the definition of Do_N, we easily know that

$$(Do_N)_{j,m}(x) = 0, \qquad \text{for } x \in \mathcal{K}_j.$$

Therefore, we have

$$\sum_{k\in\mathbb{Z}} \alpha_{j+1,k}(DD_N)_{j+1,k}(x) = \sum_{k\in\mathbb{Z}} \alpha_{j,k}(DD_N)_{j,k}(x), \qquad \text{for } x \in \mathcal{K}_j.$$

By taking $x = \dfrac{l}{2^j}$, we can derive that

$$\sqrt{2^{j+1}} \sum_{k\in\mathbb{Z}} \alpha_{j+1,k} DD_N\left(2l-k\right) = \sqrt{2^j} \sum_{k\in\mathbb{Z}} \alpha_{j,k} DD_N(l-k).$$

From the construction of DD_N, we have

$$\alpha_{j,k} = \sqrt{2}\alpha_{j+1,2k}.$$

Let $x_0 \in \mathcal{M}_j$, then $x_0 = \dfrac{k+1/2}{2^j}$ for some $k \in \mathbb{Z}$. Substituting x_0 into Eq (3.3.3) and Eq (3.3.4), and using the interpolating property of DD_N, we obtain

$$\sqrt{2}\alpha_{j+1,2k+1} = 2^{-j/2} \sum_{m\in\mathbb{Z}} \alpha_{j,m} (DD_N)_{j,m} \left(\frac{k+1/2}{2^j}\right) + 2\beta_{j,k}.$$

Hence, we achieve the following lemma.

Lemma 3.3.1. *Every $f \in V_{j+1}$ can be represented in terms of Eq* (3.3.3) *with*

$$\alpha_{j,k} = 2^{-j/2} f\left(\frac{k}{2^j}\right), \;\; \beta_{j,k} = \frac{2^{-j/2}}{2}\left(f\left(\frac{k+1/2}{2^j}\right) - (P_j f)\left(\frac{k+1/2}{2^j}\right)\right).$$

This lemma shows that the wavelet coefficients $\beta_{j,k}$'s represent the difference of f and its approximation $P_j f$ in V_j.

Now we discuss the symbols of the family of biorthogonal scaling functions and wavelets. Let h and g be the filters of DD_N and Do_N respectively and $\mathbf{h}$ and $\mathbf{g}$ be their corresponding symbols. Since DD_N is interpolating, it is easy to check from its scaling equation that $h_{2k} = \dfrac{\delta_{k,0}}{\sqrt{2}}$, for all $k \in \mathbb{Z}$. In the

symbol form,

$$\begin{aligned}\mathbf{h}(\omega)+\mathbf{h}(\omega+\pi) &= \frac{1}{\sqrt{2}}\sum_{k\in\mathbb{Z}} h_k e^{-iwk} + \frac{1}{\sqrt{2}}\sum_{k\in\mathbb{Z}} h_k(-1)^k e^{-iwk} \\ &= \sqrt{2}\sum_{k\in\mathbb{Z}} h_{2k} e^{-i2wk} \\ &= \sqrt{2}h_0 \\ &= 1.\end{aligned}$$

Thus we have

$$\mathbf{h}(\omega)+\mathbf{h}(\omega+\pi)=1, \quad \omega\in\mathbb{R}. \tag{3.3.5}$$

Since $Do_N(x) = 2DD_N(2x-1)$, for $x\in\mathbb{R}$, we know that $g_k = \sqrt{2}\delta_{k,1}$. Therefore, $\mathbf{g}(\omega) = e^{-i\omega}$, $w\in\mathbb{R}$. The biorthogonal condition Eq (3.2.10) is rewritten in a matrix form :

$$\widetilde{M}(\omega)\, M^*(\omega) = I, \tag{3.3.6}$$

where $M, \widetilde{M} \in \mathbb{C}^{2\times 2}$ written by

$$M(\omega) = \begin{bmatrix}\mathbf{h}(\omega) & \mathbf{h}(\omega+\pi)\\ \mathbf{g}(\omega) & \mathbf{g}(\omega+\pi)\end{bmatrix} \quad \text{and} \quad \widetilde{M}(\omega) = \begin{bmatrix}\widetilde{\mathbf{h}}(\omega) & \widetilde{\mathbf{h}}(\omega+\pi)\\ \widetilde{\mathbf{g}}(\omega) & \widetilde{\mathbf{g}}(\omega+\pi)\end{bmatrix},$$

and I is the 2×2 identity matrix. The upper script $*$ means *conjugate transpose* of a complex matrix.

Our goal is to define the symbols $\widetilde{\mathbf{h}}$ and $\widetilde{\mathbf{g}}$ satisfying Eq (3.3.6). Since

$$\widetilde{M}(\omega) = (M^*(\omega))^{-1},$$

there is nothing but computing the inverse of 2×2 matrix. Using the Eq

(3.3.5), $\mathbf{h}$ and $\mathbf{g}$, one can easily derive that

$$\begin{aligned}
(M^*(\omega))^{-1} &= \begin{bmatrix} \overline{\mathbf{h}(\omega)} & \overline{\mathbf{g}(\omega)} \\ \overline{\mathbf{h}(\omega+\pi)} & \overline{\mathbf{g}(\omega+\pi)} \end{bmatrix}^{-1} \\
&= \frac{1}{\overline{\mathbf{h}(\omega)\mathbf{g}(\omega+\pi) - \mathbf{h}(\omega+\pi)\mathbf{g}(\omega)}} \begin{bmatrix} \overline{\mathbf{g}(\omega+\pi)} & -\overline{\mathbf{g}(\omega)} \\ -\overline{\mathbf{h}(\omega+\pi)} & \overline{\mathbf{h}(\omega)} \end{bmatrix} \\
&= \frac{1}{-\mathbf{g}(-\omega)} \begin{bmatrix} -\mathbf{g}(-\omega) & -\mathbf{g}(-\omega) \\ -\overline{\mathbf{h}(\omega+\pi)} & \overline{\mathbf{h}(\omega)} \end{bmatrix} \\
&= \begin{bmatrix} 1 & 1 \\ \overline{\mathbf{h}(\omega+\pi)}\mathbf{g}(\omega) & -\overline{\mathbf{h}(\omega)}\mathbf{g}(\omega) \end{bmatrix} \\
&= \begin{bmatrix} 1 & 1 \\ \overline{\mathbf{h}(\omega+\pi)}\mathbf{g}(\omega) & \overline{\mathbf{h}(\omega)}\mathbf{g}(\omega+\pi) \end{bmatrix} \\
&= \begin{bmatrix} 1 & 1 \\ \overline{\mathbf{h}(\omega+\pi)}\mathbf{g}(\omega) & \overline{\mathbf{h}(\omega+\pi)}\mathbf{g}(\omega+\pi) \end{bmatrix}.
\end{aligned}$$

Therefore, we can choose

$$\widetilde{\mathbf{h}}(\omega) = 1 \quad \text{and} \quad \widetilde{\mathbf{g}}(\omega) = e^{-i\omega}\overline{\mathbf{h}(\omega+\pi)},$$

to impose a biorthogonality condition on the family $\{\mathbf{h}, \mathbf{g}, \widetilde{\mathbf{h}}, \widetilde{\mathbf{g}}\}$. Thus, the dual scaling function $\widetilde{\phi}$ is the Dirac function at the origin and the dual wavelet $\widetilde{\psi}$ is a linear combination of the shifted Dirac delta functions. The biorthogonal family $\{DD_N, Do_N, \widetilde{\phi}, \widetilde{\psi}\}$ with its family of symbols $\{\mathbf{h}, \mathbf{g}, \widetilde{\mathbf{h}}, \widetilde{\mathbf{g}}\}$ is called *Donoho wavelets family* [49].

Remark 3.3.2. The Donoho wavelets family has some disadvantages.

1. Do_N does not have vanishing moments. Thus, it does not satisfy the admissible condition, which means it cannot generate a Riesz basis of $L^2(\mathbb{R})$.

2. The duals are not $L^2(\mathbb{R})$ functions.

W. Sweldens has introduced an elegant technique called a *lifting scheme* [49], [50], which can be used to construct better families of biorthogonal

scaling function and wavelets overcoming these demerits of Donoho wavelets family.

3.3.2 Lifting Scheme

In this section, we are supposed to introduce the lifting scheme devised by W. Sweldens [49] [50]. We start with the following lemma.

Lemma 3.3.3. *Assume that a family of biorthogonal symbols* $\{\mathbf{h}, \mathbf{g}^0, \widetilde{\mathbf{h}}^0, \widetilde{\mathbf{g}}\}$ *is given. Then, define the new family of symbols* $\{\mathbf{h}, \mathbf{g}, \widetilde{\mathbf{h}}, \widetilde{\mathbf{g}}\}$ *by*

$$\begin{aligned} \mathbf{g}(\omega) &= \mathbf{g}^0(\omega) - \mathbf{h}(\omega)s(2\omega), \\ \widetilde{\mathbf{h}}(\omega) &= \widetilde{\mathbf{h}}^0(\omega) + \widetilde{\mathbf{g}}(\omega)\overline{s(2\omega)}, \end{aligned} \tag{3.3.7}$$

for $\omega \in \mathbb{R}$*, where* s *is a trigonometric polynomial. Then the new family of symbols still has the biorthogonality property.*

Proof. Our aim is to check the new family of symbols satisfies Eq (3.3.6). Let $M^0, \widetilde{M}^0, M$ and $\widetilde{M}$ be the following 2×2 complex matrices. i.e.,

$$M^0(\omega) = \begin{bmatrix} \mathbf{h}(\omega) & \mathbf{h}(\omega+\pi) \\ \mathbf{g}^0(\omega) & \mathbf{g}^0(\omega+\pi) \end{bmatrix}, \qquad \widetilde{M}^0(\omega) = \begin{bmatrix} \widetilde{\mathbf{h}}(\omega) & \widetilde{\mathbf{h}}^0(\omega+\pi) \\ \widetilde{\mathbf{g}}(\omega) & \widetilde{\mathbf{g}}(\omega+\pi) \end{bmatrix},$$
$$M(\omega) = \begin{bmatrix} \mathbf{h}(\omega) & \mathbf{h}(\omega+\pi) \\ \mathbf{g}(\omega) & \mathbf{g}(\omega+\pi) \end{bmatrix}, \qquad \widetilde{M}(\omega) = \begin{bmatrix} \widetilde{\mathbf{h}}(\omega) & \widetilde{\mathbf{h}}(\omega+\pi) \\ \widetilde{\mathbf{g}}(\omega) & \widetilde{\mathbf{g}}(\omega+\pi) \end{bmatrix}.$$

From Eq (3.3.7), we know that

$$\widetilde{M}(\omega) = S_1(\omega)\widetilde{M}^0(\omega) \quad \text{and} \quad M(\omega) = S_2(\omega)M^0(\omega),$$

where $S_1(\omega)$ and $S_2(\omega)$ are 2×2 matrices defined by

$$S_1(\omega) = \begin{bmatrix} 1 & \overline{s(2\omega)} \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad S_2(\omega) = \begin{bmatrix} 1 & 0 \\ -s(2\omega) & 1 \end{bmatrix}.$$

For simplicity, the argument ω is supposed to be omitted in the following

computations,

$$\widetilde{M}M^* = S_1\widetilde{M}^0(M^0)^*S_2^* = S_1IS_2^* = S_1S_2^* = I.$$

Therefore, the new family of symbols is also biorthogonal. □

Next, we observe how these change of symbols affect the corresponding functions. One can notice that the scaling function ϕ^0 does not change, namely $\phi = \phi^0$, since its symbol $\mathbf{h}^0$ does not change. On the other hand, the rest of functions except ϕ^0 change. In order to make the situation clear, we consider the scaling equations of these three functions $\psi, \widetilde{\phi}, \widetilde{\psi}$:

$$\begin{aligned}\widehat{\psi} &= \mathbf{g}\left(\frac{\omega}{2}\right)\widehat{\phi}\left(\frac{\omega}{2}\right)\\ &= \mathbf{g}^0\left(\frac{\omega}{2}\right)\widehat{\phi}^0\left(\frac{\omega}{2}\right) - s(\omega)\mathbf{h}^0\left(\frac{\omega}{2}\right)\widehat{\phi}^0\left(\frac{\omega}{2}\right)\\ &= \widehat{\psi}^0(\omega) - s(\omega)\widehat{\phi}^0(\omega).\end{aligned}$$

Denote the trigonometric polynomial $s(\omega)$ by

$$s(\omega) = \sum_{k\in\mathbb{Z}} s_k e^{-iwk}, \quad s_k \in \mathbb{R}.$$

Then by applying the inverse Fourier transformation, we obtain

$$\psi(x) = \sqrt{2}\sum_{k\in\mathbb{Z}} g_k^0\phi^0(2x-k) - \sum_{k\in\mathbb{Z}} s_k\phi^0(x-k).$$

In a similar manner, we also have

$$\begin{aligned}\widehat{\widetilde{\phi}}(\omega) &= \widetilde{\mathbf{h}}\left(\frac{\omega}{2}\right)\widehat{\widetilde{\phi}}\left(\frac{\omega}{2}\right)\\ &= \widetilde{\mathbf{h}}^0\left(\frac{\omega}{2}\right)\widehat{\widetilde{\phi}}\left(\frac{\omega}{2}\right) + \overline{s(\omega)}\widetilde{\mathbf{g}}^0\left(\frac{\omega}{2}\right)\widehat{\widetilde{\phi}}\left(\frac{\omega}{2}\right)\\ &= \widetilde{\mathbf{h}}^0\left(\frac{\omega}{2}\right)\widehat{\widetilde{\phi}}\left(\frac{\omega}{2}\right) + \overline{s(\omega)}\widehat{\widetilde{\psi}}(\omega).\end{aligned}$$

By the inverse Fourier transformation, we obtain

$$\widetilde{\phi}(x) = \sqrt{2}\sum_{k\in\mathbb{Z}} \widetilde{\mathbf{h}}_k^0 \widetilde{\phi}(2x-k) + \sum_{k\in\mathbb{Z}} s_{-k}\widetilde{\psi}(x-k).$$

We omit the derivation of the formula for $\widetilde{\psi}$. Finally, we can achieve the following theorem.

Theorem 3.3.4. *Suppose that we have a family of biorthogonal scaling functions and wavelets $\{\phi^0, \psi^0, \widetilde{\phi}^0, \widetilde{\psi}^0\}$. Then a new family $\{\phi, \psi, \widetilde{\phi}, \widetilde{\psi}\}$, defined by*

$$\begin{aligned}
\phi(x) &= \phi^0(x),\\
\psi(x) &= \psi^0(x) - \sum_{k\in\mathbb{Z}} s_k \phi(x-k),\\
\widetilde{\phi}(x) &= \sqrt{2}\sum_{k\in\mathbb{Z}} \widetilde{h}_k^0 \widetilde{\phi}(2x-k) + \sum_{k\in\mathbb{Z}} s_{-k}\widetilde{\psi}(x-k),\\
\widetilde{\psi}(x) &= \sqrt{2}\sum_{k\in\mathbb{Z}} \widetilde{g}_k \widetilde{\phi}(2x-k),
\end{aligned}$$

is biorthogonal for any sequence of real numbers $s_k \in \mathbb{R}$.

Remark 3.3.5.

1. This process of obtaining a new family of biorthogonal scaling functions and wavelets is called a *lifting*.

2. Assume we have an initial family $\{\mathbf{h}^0, \mathbf{g}, \widetilde{\mathbf{h}}, \widetilde{\mathbf{g}}^0\}$ of biorthogonal symbols. Then we can also construct a new family $\{\mathbf{h}, \mathbf{g}, \widetilde{\mathbf{h}}, \widetilde{\mathbf{g}}\}$ of biorthogonal symbols defined by

$$\begin{aligned}
\mathbf{h}(\omega) &= \mathbf{h}^0(\omega) + \mathbf{g}(\omega)\widetilde{s}(2\omega),\\
\widetilde{\mathbf{g}}(\omega) &= \widetilde{\mathbf{g}}^0(\omega) - \widetilde{\mathbf{h}}(\omega)\overline{\widetilde{s}(2\omega)},
\end{aligned}$$

   where $\widetilde{s}$ is a trigonometric polynomial. We call this process a *dual lifting*.

3. The choice of trigonometric polynomials $s(\widetilde{s})$ in lifting(dual lifting) processes plays an important role in improving the properties of the scaling functions and wavelets.

We introduce an interesting example to demonstrate the lifting process.

Example 3.3.6. Consider a family of biorthogonal symbols defined by

$$\mathbf{h}^0(\omega) = \frac{1}{2} \quad \text{and} \quad \widetilde{\mathbf{h}}(\omega) = 1,$$
$$\mathbf{g}(\omega) = e^{-i\omega} \quad \text{and} \quad \widetilde{\mathbf{g}}^0(\omega) = \frac{e^{-i\omega}}{2}.$$

The above family is called *Lazy wavelet* which is the simplest example of wavelet.

We assume that a family $\{\mathbf{h}, \mathbf{g}, \widetilde{\mathbf{h}}, \widetilde{\mathbf{g}}\}$ is a Donoho wavelet family. It follows that $\mathbf{h}$ is a symbol of DD_N and

$$\mathbf{g}(\omega) = e^{-i\omega}, \widetilde{\mathbf{h}}(\omega) = 1 \quad \text{and} \quad \widetilde{\mathbf{g}}(\omega) = e^{-i\omega}\overline{\mathbf{h}(\omega + \pi)}.$$

We will find certain relations between these two families $\{\mathbf{h}^0, \mathbf{g}, \widetilde{\mathbf{h}}, \widetilde{\mathbf{g}}^0\}$ and $\{\mathbf{h}, \mathbf{g}, \widetilde{\mathbf{h}}, \widetilde{\mathbf{g}}\}$.

Because $\mathbf{h}$ is a symbol of an ISF, we know that $h_k = \dfrac{DD_N(\frac{k}{2})}{\sqrt{2}}$ as well as the support of h_k is finite. Hence, there exists a trigonometric polynomial $\widetilde{s}(\omega)$ such that

$$\widetilde{s}(2\omega) = e^{i\omega}\left(\mathbf{h}(\omega) - \frac{1}{2}\right).$$

Then we have

$$\begin{aligned}\mathbf{h}(\omega) &= \frac{1}{2} + e^{-i\omega}\widetilde{s}(2\omega) \\ &= \mathbf{h}^0(\omega) + \mathbf{g}(\omega)\widetilde{s}(2\omega).\end{aligned}$$

Furthermore, the equation $\mathbf{h}(\omega) + \mathbf{h}(\omega + \pi) = 1$ tells us that

$$\begin{aligned}
\widetilde{\mathbf{g}}(\omega) &= e^{-i\omega}\overline{\mathbf{h}(\omega+\pi)} \\
&= e^{-i\omega}\left(\overline{1 - \mathbf{h}(\omega)}\right) \\
&= \frac{1}{2}e^{-i\omega} + e^{-i\omega}\left(\overline{\frac{1}{2} - \mathbf{h}(\omega)}\right) \\
&= \widetilde{\mathbf{g}}^0(\omega) - \widetilde{\mathbf{h}}(\omega)\overline{\widetilde{s}(2\omega)}.
\end{aligned}$$

Therefore, we can conclude that the family of Donoho wavelets can be obtained by carrying out a dual lifting scheme on the family of Lazy wavelets.

3.3.3 Lifting Donoho wavelets family

Now we will continue to lift the family of Donoho wavelets in order to improve the properties of wavelet functions. We start with the family of symbols of Donoho wavelets. Suppose that a family $\{\mathbf{h}, \mathbf{g}, \widetilde{\mathbf{h}}, \widetilde{\mathbf{g}}\}$ is a Donoho family of order N, that is :

$$\begin{aligned}
h_k = \frac{DD_N(\frac{k}{2})}{\sqrt{2}} \quad &\text{and} \quad \mathbf{g}(\omega) = e^{-i\omega} \\
\widetilde{\mathbf{h}}(\omega) = 1 \quad &\text{and} \quad \widetilde{\mathbf{g}}(\omega) = e^{-i\omega}\left(\overline{\mathbf{h}(\omega+\pi)}\right).
\end{aligned}$$

Let us consider the following new family $\{\mathbf{h}, \mathbf{g}^l, \widetilde{\mathbf{h}}^l, \widetilde{\mathbf{g}}\}$ lifted from the Donoho family :

$$\begin{aligned}
\mathbf{g}^l(\omega) &= e^{-i\omega} - \mathbf{h}(\omega)s(2\omega) \\
\widetilde{\mathbf{h}}^l(\omega) &= 1 + e^{-i\omega}\overline{\mathbf{h}(\omega+\pi)s(2\omega)},
\end{aligned}$$

where s is a trigonometric polynomial. Denoting $s(\omega) = \sum_{k\in\mathbb{Z}} s_k e^{-ik\omega}$, we know from the theorem (3.3.4) that the lifted wavelet ψ^l is written as

$$\begin{aligned}\psi^l(x) &= Do_N(x) - \sum_{k\in\mathbb{Z}} s_k DD_N(x-k) \\ &= 2DD_N(2x-1) - \sum_{k\in\mathbb{Z}} s_k DD_N(x-k).\end{aligned}$$

Now, our main goal is to choose the coefficients s_k so that ψ^l has $2\widetilde{N}$ vanishing moments. From the Example (3.3.6), we already know that dual lifting induced by the trigonometric polynomial $\widetilde{s}(\omega)$ with

$$\widetilde{s}(2\omega) = e^{i\omega}\left(\mathbf{h}(\omega) - \frac{1}{2}\right),$$

constructs Donoho wavelets from the Lazy wavelets. By the fact that $h_k = \dfrac{DD_N(\frac{k}{2})}{\sqrt{2}}$, we can easily check that the coefficients of $\widetilde{s}$ is half of the $2N$-points Lagrangian interpolation weights :

$$\begin{aligned}\widetilde{s}(2\omega) &= e^{i\omega}\left(\frac{1}{\sqrt{2}}\sum_{k\in\mathbb{Z}} h_k e^{-i\omega k} - \frac{1}{2}\right) \\ &= e^{i\omega}\left(\frac{1}{2}\sum_{k\in\mathbb{Z}} DD_N\left(\frac{k}{2}\right)e^{-i\omega k} - \frac{1}{2}\right) \\ &= \frac{1}{2}\sum_{k\in\mathbb{Z}} DD_N\left(k+\frac{1}{2}\right)e^{-2i\omega k} \\ &= \frac{1}{2}\sum_{k=-2N+1}^{2N-1} DD_N\left(k+\frac{1}{2}\right)e^{-2i\omega k}.\end{aligned}$$

Therefore, if we denote $\widetilde{s}(\omega) = \sum_{k\in\mathbb{Z}} \widetilde{s}_k e^{-i\omega k}$, we have

$$\widetilde{s}_k = \frac{1}{2} DD_N\left(k+\frac{1}{2}\right). \tag{3.3.8}$$

Finally, we arrive at the very important theorem. The following theorem tells us that the lifted Donoho wavelet can be constructed so that the disadvantages mentioned in Remark (3.3.2) are resolved. W. Sweldens [49] has shown the marvellous theorem. The proof of the theorem is omitted since it is too technical.

Theorem 3.3.7. *Let $N, \widetilde{N} \in \mathbb{N}$ be given. Suppose that $\mathbf{h}^{\widetilde{N}}$ is the symbol of $DD_{\widetilde{N}}$. We define the trigonometric polynomial $\widetilde{s}^{\widetilde{N}}$ as*

$$\widetilde{s}^{\widetilde{N}}(\omega) = e^{i\omega/2}\left(\mathbf{h}^{\widetilde{N}}(\omega/2) - 1/2\right).$$

If $\widetilde{N} \leq N$, the lifted Donoho wavelet family with

$$s(\omega) = 2\widetilde{s}^{\widetilde{N}}(-\omega)$$

results in the shortest wavelet with $2\widetilde{N}$ vanishing moments.

We denote the lifted Donoho wavelet with $2N$ vanishing moments by Dl_N. In order to make the situation clear, we consider a simple example for the case $N = \widetilde{N}$. Because we already have Equation (3.3.8), it is trivial that

$$s(\omega) = 2\widetilde{s}(-\omega) = \sum_{k\in\mathbb{Z}} 2\widetilde{s}_{-k}e^{-i\omega k}.$$

By the symmetric property of DD_N, we finally have

$$s_k = 2\widetilde{s}_{-k} = DD_N\left(k - \frac{1}{2}\right).$$

3.3.4 The Lifted interpolating wavelet transform

Suppose $N \in \mathbb{N}$ is fixed. Let $\{DD_N, Do_N, \widetilde{\phi}, \widetilde{\psi}\}$ be a family of Donoho wavelets and $\{DD_N, Dl_N\}$ be a family lifted from the Donoho wavelets family. Assume that $s(\omega)$ is the trigonometric polynomial mentioned in Theorem 3.3.7. And let $\{\mathbf{h}, \mathbf{g}, \widetilde{\mathbf{h}}, \widetilde{\mathbf{g}}\}$ and $\{\mathbf{h}, \mathbf{g}^l, \widetilde{\mathbf{h}}^l, \widetilde{\mathbf{g}}\}$ be the corresponding symbols of Donoho wavelets and the lifted Donoho wavelet, respectively.

A main goal of this subsection is to derive formula of the wavelet transformation with the lifted Donoho wavelets. We start with applying Theorem 3.3.4 to Donoho wavelets :

$$\begin{aligned}\widetilde{\phi}^l(x) &= \sqrt{2}\sum_{l\in\mathbb{Z}}\widetilde{h}_l\widetilde{\phi}^l(2x-l)+\sum_{l\in\mathbb{Z}}s_{-l}\widetilde{\psi}^l(x-l),\\ \widetilde{\psi}^l(x) &= \sqrt{2}\sum_{l\in\mathbb{Z}}\widetilde{g}_l\widetilde{\phi}^l(2x-l).\end{aligned} \tag{3.3.9}$$

Since $\widetilde{\mathbf{h}}(\omega)=1$, we know that

$$\widetilde{h}_l=\sqrt{2}\delta_{l,0}.$$

Because

$$\begin{aligned}\widetilde{\mathbf{g}}(\omega) &= e^{-i\omega}\overline{\mathbf{h}(\omega+\pi)}\\ &= e^{-i\omega}\left(\overline{1/2-e^{-i\omega}\widetilde{s}(2\omega)}\right)\\ &= e^{-i\omega}(1/2-e^{i\omega}\widetilde{s}(-2\omega))\\ &= \frac{1}{2}e^{-i\omega}-\sum_{l\in\mathbb{Z}}\widetilde{s}_{-l}e^{-i2\omega l},\end{aligned}$$

we obtain that

$$\begin{aligned}\widetilde{g}_{2l+1} &= \frac{1}{\sqrt{2}}\delta_{l,0},\\ \widetilde{g}_{2l} &= -\sqrt{2}\widetilde{s}_{-l},\end{aligned}$$

for $l\in\mathbb{Z}$.

By modifying Equation (3.3.9), we obtain

$$\begin{aligned}\widetilde{\phi}^l_{j,k}(x) &= \sum_{l\in\mathbb{Z}}\widetilde{h}_l\widetilde{\phi}^l_{j+1,2k+l}(x)+\sum_{l\in\mathbb{Z}}s_{-l}\widetilde{\psi}^l_{j,k+l}(x),\\ \widetilde{\psi}^l_{j,k}(x) &= \sum_{l\in\mathbb{Z}}\widetilde{g}_l\widetilde{\phi}^l_{j+1,2k+l}(x).\end{aligned}$$

Using the general forms of $\widetilde{h}_l$ and $\widetilde{g}_l$ computed above, we have

$$\begin{aligned}\widetilde{\phi}^l_{j,k}(x) &= \sqrt{2}\widetilde{\phi}^l_{j+1,2k}(x) + \sum_{l\in\mathbb{Z}} s_{-l}\widetilde{\psi}^l_{j,k+l}(x),\\ \widetilde{\psi}^l_{j,k}(x) &= \frac{1}{\sqrt{2}}\widetilde{\phi}^l_{j+1,2k+1}(x) - \sqrt{2}\sum_{l\in\mathbb{Z}} \widetilde{s}_{-l}\widetilde{\phi}^l_{j+1,2k+2l}(x).\end{aligned}$$

Taking the inner product on both sides, for $f \in L^2(\mathbb{R})$, we get the following formula,

$$\begin{aligned}\alpha_{j,k} &= \sqrt{2}\alpha_{j+1,2k} + \sum_{l\in\mathbb{Z}} s_{-l}\beta_{j,k+1},\\ \beta_{j,k} &= \frac{1}{\sqrt{2}}\left(\alpha_{j+1,2k+1} - \sum_{l\in\mathbb{Z}} 2\widetilde{s}_{-l}\alpha_{j+1,2k+2l}\right).\end{aligned}$$

For convenience, we define a normalized coefficients $c_{j,k}$ and $d_{j,k}$ as follows :

$$c_{j,k} = \sqrt{2^j}\alpha_{j,k} \quad \text{and} \quad d_{j,k} = \sqrt{2^j}\beta_{j,k}.$$

Then, we eventually obtain the normalized wavelet transform :

$$\begin{aligned}c_{j,k} &= c_{j+1,2k} + \sum_{l\in\mathbb{Z}} s_{-l}d_{j,k+1},\\ d_{j,k} &= \frac{1}{2}\left(c_{j+1,2k+1} - \sum_{l\in\mathbb{Z}} 2\widetilde{s}_{-l}c_{j+1,2k+2l}\right).\end{aligned} \tag{3.3.10}$$

We should discuss an initializing stage. Assume that $f \in L^2(\mathbb{R})$, and j is the highest resolution level. Then we approximate f on V_j by the projection,

$$\begin{aligned}P_j f(x) &= \sum_{k\in\mathbb{Z}} \alpha_{j,k}(DD_N)_{j,k}(x)\\ &= \sum_{k\in\mathbb{Z}} c_{j,k}DD_N(2^j x - k).\end{aligned}$$

By the interpolating property of DD_N, we have

$$c_{j,k} = f\left(\frac{k}{2^j}\right).$$

Furthermore, families of one-dimensional biorthogonal wavelets are easily extended to multi-dimensional spaces. We consider *separable wavelet bases* [45]. It is well known that an one-dimensional MRA $\{V_j\}_{j\in\mathbb{Z}}$ with biorthgonal wavelets $\{\phi, \psi, \widetilde{\phi}, \widetilde{\psi}\}$ induces the two-dimensional MRA $\{V_j^2 = V_j \otimes V_j\}_{j\in\mathbb{Z}}$ with biorthogonal wavelets $\{\Phi, \Psi^\mu, \widetilde{\Phi}, \widetilde{\Psi}^\mu\}$ defined as

$$\Phi_{j,m,n}(x,y) = \phi_{j,m}(x)\phi_{j,n}(y),$$
$$\Psi^\mu_{j,m,n}(x,y) = \begin{cases} \psi_{j,m}(x)\phi_{j+1,2n}(y), & \mu = 1, \\ \phi_{j+1,2m}(x)\psi_{j,n}(y), & \mu = 2, \\ \psi_{j,m}(x)\psi_{j,n}(y), & \mu = 3, \end{cases}$$

and

$$\widetilde{\Phi}_{j,m,n}(x,y) = \widetilde{\phi}_{j,m}(x)\widetilde{\phi}_{j,n}(y),$$
$$\widetilde{\Psi}^\mu_{j,m,n}(x,y) = \begin{cases} \widetilde{\psi}_{j,m}(x)\widetilde{\phi}_{j+1,2n}(y), & \mu = 1, \\ \widetilde{\phi}_{j+1,2m}(x)\widetilde{\psi}_{j,n}(y), & \mu = 2, \\ \widetilde{\psi}_{j,m}(x)\widetilde{\psi}_{j,n}(y), & \mu = 3. \end{cases}$$

Therefore, the two-dimensional wavelet transform can be derived as follows :

$$\begin{aligned}
\widetilde{\Phi}_{j,m,n}(x,y) &= \widetilde{\phi}_{j,m}(x)\widetilde{\phi}_{j,n}(y) \\
&= \left(\sqrt{2}\widetilde{\phi}_{j+1,2m}(x) + \sum_{l\in\mathbb{Z}} s_{-l}\widetilde{\psi}_{j,m+l}(x)\right)\left(\sqrt{2}\widetilde{\phi}_{j+1,2n}(y) + \sum_{l'\in\mathbb{Z}} s_{-l'}\widetilde{\psi}_{j,n+l'}(y)\right) \\
&= 2\widetilde{\phi}_{j+1,2m}(x)\widetilde{\phi}_{j+1,2n}(y) + \sqrt{2}\sum_{l\in\mathbb{Z}} s_{-l}\widetilde{\psi}_{j,m+l}(x)\widetilde{\phi}_{j+1,2n}(y) \\
&\quad + \sqrt{2}\sum_{l'\in\mathbb{Z}} s_{-l'}\widetilde{\phi}_{j+1,2m}(x)\widetilde{\psi}_{j,n+l'}(y) + \sum_{l\in\mathbb{Z}}\sum_{l'\in\mathbb{Z}} s_{-l}s_{-l'}\widetilde{\psi}_{j,m+l}(x)\widetilde{\psi}_{j,n+l'}(y) \\
&= 2\Phi_{j+1,m,n}(x,y) + \sqrt{2}\sum_{l\in\mathbb{Z}} s_{-l}\Psi^1_{j,m+l,n}(x,y) \\
&\quad + \sqrt{2}\sum_{l'\in\mathbb{Z}} s_{-l'}\Psi^2_{j,m,n+l'}(x,y) + \sum_{l\in\mathbb{Z}}\sum_{l'\in\mathbb{Z}} s_{-l}s_{-l'}\Psi^3_{j,m+l,n+l'}(x,y),
\end{aligned}$$

and

$$\begin{aligned}
\widetilde{\Psi}^1_{j,m,n}(x,y) &= \widetilde{\psi}_{j,m}(x)\widetilde{\phi}_{j+1,2n}(y) \\
&= \left(\frac{1}{\sqrt{2}}\widetilde{\phi}_{j+1,2m+1}(x) - \sqrt{2}\sum_{l\in\mathbb{Z}} \widetilde{s}_{-l}\widetilde{\phi}_{j+1,2m+2l}(x)\right)\widetilde{\phi}_{j+1,2n}(y) \\
&= \frac{1}{\sqrt{2}}\left(\widetilde{\Phi}_{j+1,2m+1,2n}(x,y) - \sum_{l\in\mathbb{Z}} 2\widetilde{s}_{-l}\widetilde{\Phi}_{j+1,2m+2l,2n}(x,y)\right),
\end{aligned}$$

similarly

$$\widetilde{\Psi}^2_{j,m,n}(x,y) = \frac{1}{\sqrt{2}}\left(\widetilde{\Phi}_{j+1,2m,2n+1}(x,y) - \sum_{l\in\mathbb{Z}} 2\widetilde{s}_{-l}\widetilde{\Phi}_{j+1,2m,2n+2l}(x,y)\right),$$

and

$$\begin{aligned}
\widetilde{\Psi}^3_{j,m,n}(x,y) = \frac{1}{2}\Bigg(&\widetilde{\Phi}_{j+1,2m+1,2n+1}(x,y) - \sum_{l\in\mathbb{Z}} 2\widetilde{s}_{-l}\widetilde{\Phi}_{j+1,2m+2l,2n_1} \\
&- \sum_{l'\in\mathbb{Z}} 2\widetilde{s}_{-l'}\widetilde{\Phi}_{j+1,2m+1,2n+2l'} + \sum_{l\in\mathbb{Z}}\sum_{l'\in\mathbb{Z}} (2\widetilde{s}_{-l})(2\widetilde{s}_{-l'})\widetilde{\Phi}_{j+1,2m+2l,2n+2l'}\Bigg).
\end{aligned}$$

By taking an inner product $\langle f, \cdot \rangle$ on the both sides, we have the following formula :

$$\alpha_{j,m,n} = 2\alpha_{j+1,2m,2n} + \sqrt{2}\sum_{l\in\mathbb{Z}} s_{-l}\beta^1_{j,m+l,n} + \sqrt{2}\sum_{l'} s_{-l'}\beta^2_{j,m,n+l'} + \sum_{l,l'\in\mathbb{Z}} s_{-l}s_{-l'}\beta^3_{j,m+l,n+l'},$$

$$\beta^1_{j,m,n} = \frac{1}{\sqrt{2}}\left(\alpha_{j+1,2m+1,2n} - \sum_{l\in\mathbb{Z}} 2\widetilde{s}_{-l}\alpha_{j+1,2m+2l,2n}\right),$$

$$\beta^2_{j,m,n} = \frac{1}{\sqrt{2}}\left(\alpha_{j+1,2m,2n+1} - \sum_{l\in\mathbb{Z}} 2\widetilde{s}_{-l}\alpha_{j+1,2m,2n+2l}\right),$$

$$\beta^3_{j,m,n} = \frac{1}{2}\Bigg(\alpha_{j+1,2m+1,2n+1} - \sum_{l\in\mathbb{Z}} 2\widetilde{s}_{-l}\alpha_{j+1,2m+2l,2n+1} - \sum_{l'\in\mathbb{Z}} 2\widetilde{s}_{-l'}\alpha_{j+1,2m+1,2n+2l'} + \sum_{l,l'\in\mathbb{Z}} (2\widetilde{s}_{-l})(2\widetilde{s}_{-l'})\alpha_{j+1,2m+2l,2n+2l'}\Bigg),$$

where

$$\alpha_{j,m,n} = \left\langle f, \widetilde{\Phi}_{j,m,n}\right\rangle \quad \text{and} \quad \beta^\mu_{j,m,n} = \left\langle f, \widetilde{\Psi}^\mu_{j,m,n}\right\rangle.$$

By defining the normalized coefficients $c_{j,m,n}$, $d^\mu_{j,m,n}$ as

$$c_{j,m,n} = 2^j\alpha_{j,m,n},\ d^1_{j,m,n} = 2^{j+1/2}\beta^1_{j,m,n},\ d^2_{j,m,n} = 2^{j+1/2}\beta^2_{j,m,n} \text{ and } d^3_{j,m,n} = 2^j\beta^3_{j,m,n},$$

we obtain the two-dimensional normalized wavelet transform :

$$d^1_{j,m,n} = \frac{1}{2}\left(c_{j+1,2m+1,2n} - \sum_{l\in\mathbb{Z}} 2\widetilde{s}_{-l} c_{j+1,2m+2l,2n}\right),$$

$$d^2_{j,m,n} = \frac{1}{2}\left(c_{j+1,2m,2n+1} - \sum_{l\in\mathbb{Z}} 2\widetilde{s}_{-l} c_{j+1,2m,2n+2l}\right),$$

$$d^3_{j,m,n} = \frac{1}{4}\Bigg(c_{j+1,2m+1,2n+1} - \sum_{l\in\mathbb{Z}} 2\widetilde{s}_{-l} c_{j+1,2m+2l,2n+1} - \sum_{l'\in\mathbb{Z}} 2\widetilde{s}_{-l'} c_{j+1,2m+1,2n+2l'} + \sum_{l\in\mathbb{Z}}\sum_{l'\in\mathbb{Z}} (2\widetilde{s}_{-l})(2\widetilde{s}_{-l'}) c_{j+1,2m+2l,2n+2l'}\Bigg),$$

$$c_{j,m,n} = c_{j+1,2m,2n} + \sum_{l\in\mathbb{Z}} s_{-l} d^1_{j,m+l,n} + \sum_{l'\in\mathbb{Z}} s_{-l'} d^2_{j,m,n+l'} + \sum_{l\in\mathbb{Z}}\sum_{l'\in\mathbb{Z}} s_{-l} s_{-l'} d^3_{j,m+l,n+l'}.$$

3.3.5 Compression

Let $f(x,y)$ be a function in $L^2(\mathbb{R}^2)$ and j_{min}, $j_{max} \in \mathbb{N}$ be the minimum and maximum resolution levels. Then, the function $f(x,y)$ can be written by the wavelet decomposition :

$$f(x,y) = \sum_{m,n\in\mathbb{Z}} \alpha_{j_{min},m,n}\Phi_{j_{min},m,n}(x,y) + \sum_{\mu=1}^{3}\sum_{j=j_{min}}^{\infty}\sum_{m,n\in\mathbb{Z}} \beta^{\mu}_{j,m,n}\Psi^{\mu}_{j,m,n}(x,y).$$

Then we project $f(x,y)$ onto $V^2_{j_{max}}$ to get

$$P_{V^2_{j_{max}}} f(x,y) = \sum_{m,n\in\mathbb{Z}} \alpha_{j_{min},m,n}\Phi_{j_{min},m,n}(x,y) + \sum_{\mu=1}^{3}\sum_{j=j_{min}}^{j_{max}-1}\sum_{m,n\in\mathbb{Z}} \beta^{\mu}_{j,m,n}\Psi^{\mu}_{j,m,n}(x,y).$$

Now, we compress the projected function $P_{V^2_{j_{max}}} f(x,y)$ as follows :

$$\left(P_{V^2_{j_{max}}} f(x,y)\right)_{\epsilon} = \sum_{m,n\in\mathbb{Z}} \alpha_{j_{min},m,n}\Phi_{j_{min},m,n}(x,y) + \sum_{\mu=1}^{3}\sum_{j=j_{min}}^{j_{max}-1}\sum_{m,n\in\mathbb{Z}} T^{\mu}_{\epsilon}(\beta^{\mu}_{j,m,n})\Psi^{\mu}_{j,m,n}(x,y),$$

where the thresholding function T_ϵ^μ is defined as

$$T_\epsilon^{1,2}(x) = \begin{cases} x, & \text{if} \quad |x| \geq 2^{-j-1/2}\epsilon \\ 0, & \text{otherwise.} \end{cases},$$

and

$$T_\epsilon^{3}(x) = \begin{cases} x, & \text{if} \quad |x| \geq 2^{-j}\epsilon, \\ 0, & \text{otherwise.} \end{cases}.$$

In other words, by discarding the wavelet coefficient that are negligible, the target function $f(x, y)$ can be compressed.

3.4 Wavelet-based Adaptive WENO scheme

3.4.1 Adjacent Zone

Due to the nature of hyperbolic partial differential equations that propagate the information into the adjacent region with a finite speed, a suitable narrow band is essential to gain accurate numerical solutions. Therefore, we need to add a few points to the vicinity of all active grid points. In this paper, we use the same technique presented by Vasilyev [9] in a dimension by dimension manner.

Suppose that a multi-resolution grid point (x_m^j, y_n^j) is given, where $x_m^j = m/2^j, y_n^j = n/2^j$. Then we consider the set called *Adjacent zone*, which consists of the points $(x_{m'}^{j'}, y_{n'}^{j'})$ satisfying

$$|j' - j| \leq L, \quad |2^{j'-j}m - m'| \leq M, \quad |2^{j'-j}n - n'| \leq M.$$

The parameters L, M are set to be 1 because the CFL number is usually less than 1.

3.4.2 Methodology for Spatial discretizations

The numerical flux is computed by differentiating MRA(multiresolution approximation) representation of field values in [9, 10]. However, it is well known that such fixed stencil approximation of high order accuracy is necessarily oscillatory near discontinuities which is called Gibbs phenomena [3]. When discontinuous waves are generated, artificial viscosity term is introduced to smooth out stiff behavior of field functions [52].

WENO type interpolation methods have shown excellent performance in both smooth and discontinuous regions. However the main drawback of WENO interpolation is a computational cost caused by evaluating non-linear weights and smoothness indicator involving L_1 norms of several functions. Such auxiliary operations are necessary to control some oscillations by assigning small weights to the stencil containing shocks, as if only smooth stencil is used.

Furthermore, characteristic decomposition approach that computes Roe averaged values and solves matrix inversion problem makes whole numerical simulation heavy. However, there is no need for considering elaborate treatments mentioned above, provided field function is smooth sufficiently.

Hence, we consider the two distinct methods to evaluate numerical differentiation according to the degree of smoothness of the function. The basic assumption is that large wavelet coefficients represent the details such as shock waves and oscillatory motions. Actually, it is already well known that the frequency of the discontinuous function never disappear no matter how we increase the level of resolutions [53]. Therefore, at the location where the large wavelet coefficients lie, the classical finite difference WENO scheme is applied. On the other hands, in the smooth regions, fifth order central differencing equivalent to WENO interpolation with ideal weight [6] is used in the component-wise sense(Algorithm 4).

Algorithm 4 Strategy for evaluating spatial differentiations efficiently

Consider a hyperbolic conservation system :

$$\frac{du}{dt} + \frac{d}{dx} f(u) = 0.$$

Let $P = (x_m, y_n)$ be a point on the multiresolution adaptive grid with level $j \in \{0, 1, \cdots, j_{max}\}$.

if $j = j_{max}$, **then** ▷ P is on the finest grid.

$\left(\frac{df}{dx}\right)_{m,n}$ is computed by finite difference WENO method (Algorithm 1, 2).

else if $0 \leq j < j_{max}$, **then** ▷ P lies on the coarse grid

$$\left(\frac{df}{dx}\right)_{m,n} = -\frac{1}{12}\frac{u_{m+2s,n} - u_{m-2s,n}}{s\Delta x} + \frac{2}{3}\frac{u_{m+s,n} - u_{m-s,n}}{s\Delta x},$$

where stride size $s = 2^{j_{max}-j}$, Δx is a spacing of fine grid.

end if

3.4.3 Time Integration

3rd TVD Runge-Kutta time integration evolving the function over 3 stages is used to simulate. As mentioned in [54], this time integration scheme has TVD(total variation diminishing) property and third order accuracy. The updated value u^{n+1} is obtained as a linear combination of the functions obtained from successive updates with numerical differential operator $L(\cdot)$:

$$\begin{aligned} u^{(1)} &= u^n + \Delta t L(u_n), \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}\left(u^{(1)} + \Delta t L(u^{(1)})\right), \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}\left(u^{(2)} + \Delta t L(u^{(2)})\right). \end{aligned}$$

Local time stepping technique that marches the solution along different time step size determined according to the grid resolution is introduced with AWCM(Adaptive Wavelet Collocation Method) [55], AMR(Adaptive Mesh Refinement) method [56]. Efficiency and good agreement with full grid simulations could be achieved by avoiding the strict restriction of time step [56].

The efficiency of local time stepping is considerable when the cost of spatial differential operator $L(\cdot)$ at the coarse grid is heavy. In our framework, however, on the coarse grid points $L(\cdot)$ is computed effectively by simple central differencing without characteristic decompositions. Therefore, we adhere to use the 3rd TVD Runge-Kutta time integration.

3.4.4 Conservation error and boundary treatment

The proposed scheme is not fully conservative since two different spatial approximations are used to compute derivatives at points which lie on grids having various scales in finite difference framework. It, however, must be emphasized that the conservation error doesn't affect significantly, provided the solution is smooth sufficiently [57]. Moreover, Sebastian, Shu [58] have shown that Lagrangian interpolation is sufficient to retain essential conservation, high order accuracy, essentially non-oscillatory properties at the interfaces of multi-domains. Li [59] demonstrated that the conservation error vanishes rapidly as the grid resolution gets larger through the simulation of one dimensional burger equation during very long time period.

The points located near the boundary are not compressed in order to prevent from the contamination of the solution caused by the compressed boundary informations. Such boundary portion hardly influences the total compression ratio.

3.4.5 Overall Process

Algorithm 5 Pseudo-code for evolving solution from u^n into u^{n+1}

1. Take a forward wavelet transformation to u^n with $Mask^n$.
2. Compress u^n with thresholding parameter ϵ.
3. Extend $Mask^n$ to involve the adjacent zone(Say $Mask^{n+1}$).
4. Add points required to compute high order numerical derivative to the $Mask^{n+1}$(Say $Mask^{temp}$).
5. Perform inverse wavelet transformation(interpolation) to u^n with $Mask^{temp}$.
6. Compute spatial differential operator $L(\cdot)$

   Traversing the $Mask^{n+1}$, iterate :

   Compute the numerical spatial derivative $L(u^n)$ obeying the Algorithm 4.

   Update from u^n to u^{n+1} using $L(u^n)$.

3.5 Numerical results

For all numerical tests to be illustrated below, the adaptation method that determines which grid points are used in the simulations is applied to the density field.

3.5.1 1-dimensional equations

Shock Entropy Interaction problem

The initial condition of shock wave interaction problem devised by [5] is given by

$$(\rho,\ u,\ p) = \begin{cases} (3.857143,\ 2.629369,\ 10.33333) & \text{if } x < -4 \\ \left(1 + \frac{1}{5}\sin(5x),\ 0,\ 1\right) & \text{if } x \geq -4 \end{cases}$$

with the specific heat ratio $\gamma = 1.4$. The left and right boundaries of computational domain $[-5,\ 5]$ are set to be free outflow boundary. Evolution of the physical variables are carried out until $t = 1.8$. The adaptivity parameters are given as $j_{max} = 5$, $\epsilon = 10^{-4}$ and the 30 coarsest grid points used. This example is a good indicator for verifying whether the scheme captures both smooth and discontinuous behaviors efficiently or not.

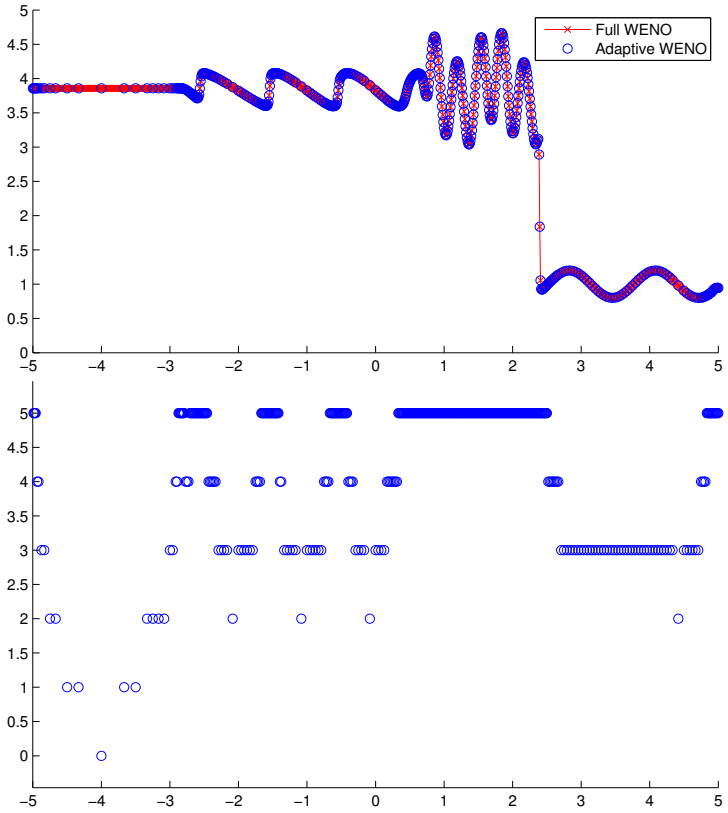


Figure 3.3: Shock entropy wave interaction. Density profiles of original WENO and the proposed method at $t = 1.8$ with 960 grid points(top). The 44% of grid points are used in the adaptive method. Level plot is shown to verify that our method detects discontinuous regions as the finest grid points(bottom).

3.5.2 2-dimensional Euler equations

2D Riemann problems

In this experiment, we simulate four two-dimensional Riemann problems proposed by [60]. This problem is one of the well known test bed for any numerical scheme solving the Euler equations. The computational domain $[0, 1] \times [0, 1]$ is divided by four uniform quadrants :

$$\text{Quad1} : [0.5, 1] \times [0.5, 1], \quad \text{Quad2} : [0, 0.5] \times [0.5, 1],$$

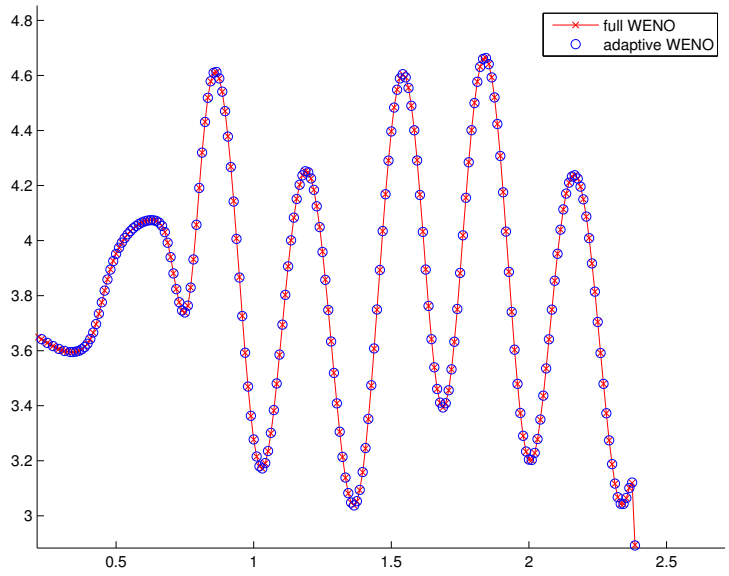


Figure 3.4: Zoomed region of Figure 3.3. The dynamic distribution of density is accurately captured.

$$\text{Quad3} : [0, 0.5] \times [0, 0.5], \quad \text{Quad4} : [0.5, 1] \times [0, 0.5]$$

The initial conditions (ρ, u, v, p) are given by Table 3.2. The outflow boundary conditions are imposed at every four computational boundaries. The shock waves, contact discontinuities and smooth rarefaction waves dynamically interact while the simulations are progressing. The numerical results are obtained with adaptive parameters chosen as $j_{\max} = 5$, $\epsilon = 10^{-3}$, and the 60×60 coarse grid is used. Good agreements with the reference solution

Table 3.2: Initial conditions for the four two-dimensional Riemann problems.

	Problem1(t = 0.25)	Problem2(t=0.25)
Quad1	(1.5000, 0.000, 0.000, 1.500)	(1.1000, 0.0000, 0.0000, 1.10)
Quad2	(0.5323, 1.206, 0.000, 0.300)	(0.5065, 0.8939, 0.0000, 0.35)
Quad3	(0.1380, 1.206, 1.206, 0.029)	(1.1000, 0.8939, 0.8939, 1.10)
Qaud4	(0.5323, 0.000, 1.206, 0.300)	(0.5065, 0.0000, 0.8939, 0.35)

	Problem3(t=0.30)	Problem4(t=0.25)
Quad1	(1, 0.75, -0.5, 1)	(0.5313, 0.0000, 0.0000, 0.4)
Quad2	(2, 0.75, 0.5, 1)	(1.0000, 0.7276, 0.0000, 1.0)
Quad3	(1, -0.75, 0.5, 1)	(0.8000, 0.0000, 0.0000, 1.0)
Qaud4	(3, -0.75, -0.5, 1)	(1.0000, 0.0000, 0.7276, 1.0)

published in [60] are seen clearly.

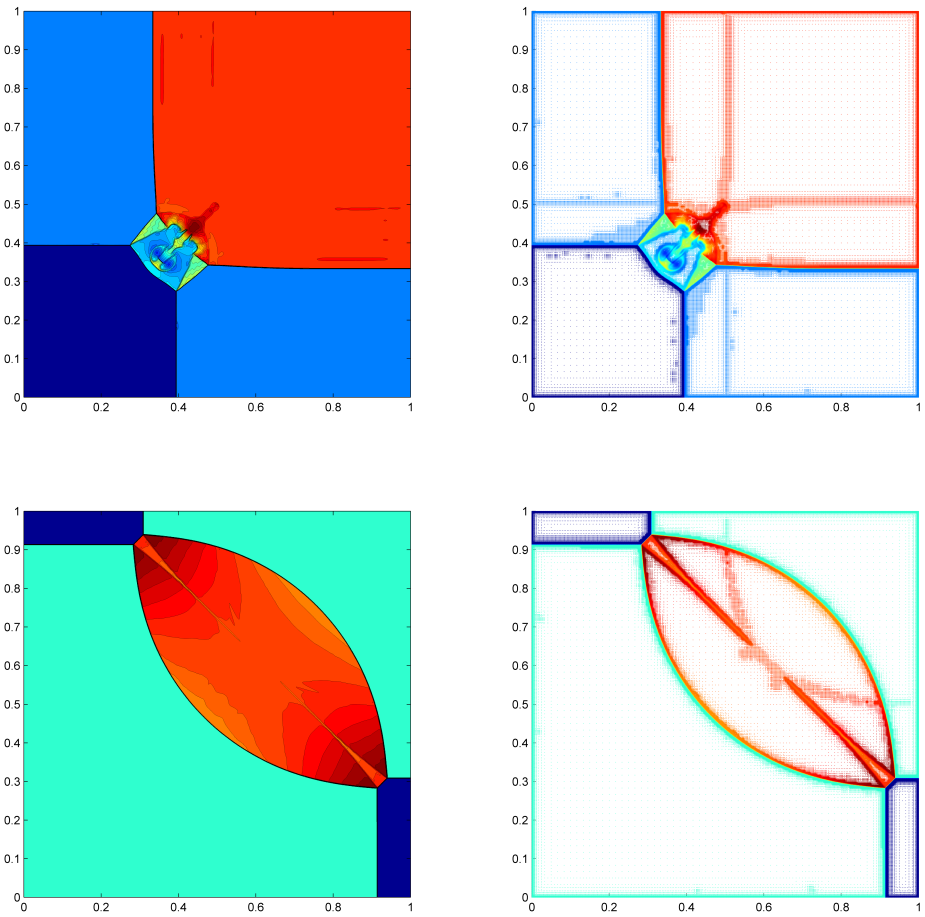


Figure 3.5: Results for two-dimensional Riemann problem 1, 2. Interpolated density contour(left) and scattered image of grid points(right) used in the simulation are shown. Compression ratios are 11.65% and 12.26% respectively.

Isentropic vortex advection

An isentropic vortex advection problem [61] is considered to assess the order of the adaptive WENO scheme. It is known that an isentropic vortex flowing along the inviscid free stream provides a good indicator to check the order

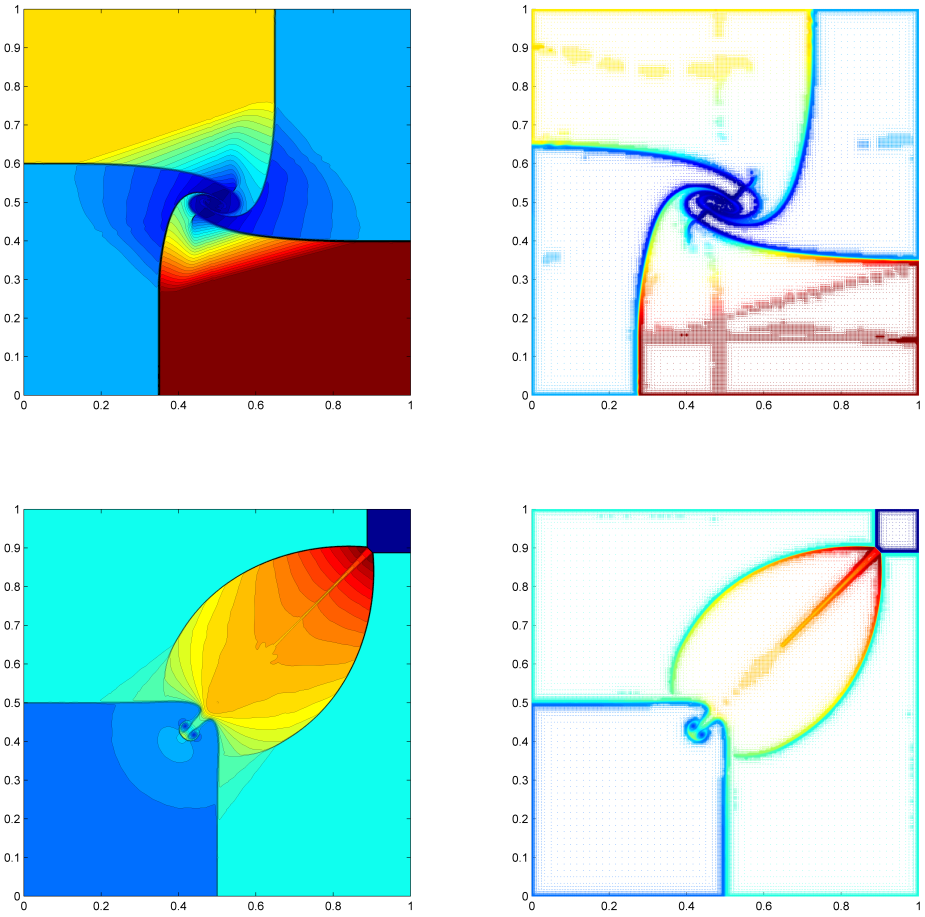


Figure 3.6: Results for two-dimensional Riemann problem 3, 4. Interpolated density contour(left) and scattered image of grid points(right) used in the simulation are shown. Compression ratios are 13.57% and 10.79% respectively.

of accuracy of schemes. Initial condition for the isentropic vortex is given by a perturbation added to a uniform mean flow

$$\rho = T^{1/(\gamma-1)} = (1+\delta T)^{1/(\gamma-1)}, \quad \rho u = \rho(1+\delta u), \quad \rho v = \rho(1+\delta v), \quad p = \rho^{\gamma},$$

where

$$(\delta u, \delta v) = \frac{\epsilon}{2\pi} e^{1-r^2}(-y, x-5), \quad \delta T = -\frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2} e^{1-r^2}, \quad r = \sqrt{(x-5)^2+y^2}.$$

The computational domain is $[0,10] \times [-5,5]$, the vortex strength ϵ is set to be 5 , $\gamma = 1.4$ and simulation is performed until $t = 10$. To exclude inconsistency of the compression ratio in the convergence test(Table 3.3), the threshold parameter ϵ is tuned so that the compression ratio stays around 65%.

The Figure 3.7 containing compressed data and interpolated data is presented to illustrate the effect of adaptivity.

Double mach reflection

The *double mach reflection* problem is a classical test problem for the Euler equations. It was designed to simulate the behavior of an oblique mach 10 shock interacting with the horizontal reflecting wedge [62]. The computational domain is $[0,4] \times [0,1]$. Initially, the oblique shock intersects with the x-axis at $x = 1/6$ with an angle of degree 60°. At the left and right boundaries in-flow and out-flow conditions are applied, respectively. The bottom

Table 3.3: Grid refinement test for isentropic vortex evaluation problem. The L_1 norms of the error of density ρ at $t = 10$ are presented to evaluate the order of the adaptive WENO scheme.

Grid size	L_1 - error	Average percent of points used	order
32×32	0.8977	70.8	none
64×64	0.0176	63.7	5.6726
96×96	0.0023	63.8	5.0189
128×128	4.7336e-04	62.7	5.4950
160×160	1.3704e-04	66.4	5.5551

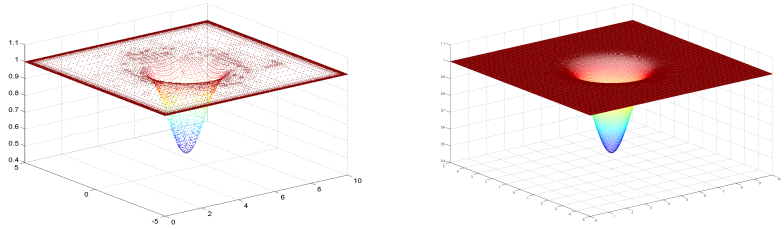


Figure 3.7: Compressed data(left) and interpolated data(right) for the density.(grid size : 320×320, compression ratio : 13%) Only compressed data are used in the process of adaptive WENO along each time steps, then the data is interpolated in the final stage(or necessary stages). The parameters used are $j_{\max} = 5$, $N_{coarse} = 21$, $\epsilon = 10^{-5}$.

boundary, starting from the intersection point $x = 1/6$, is a reflection wall. The exact solution is given on the top boundary and the bottom boundary. We set the coarsest grid to 60×15, the maximum resolution level to $j_{max} = 6$ and the threshold to $\varepsilon = 5.0 \times 10^{-3}$. The simulation is performed until $t = 0.2$. The numerical result is given in Figure 3.8, 3.9, 3.10, 3.11 and 3.12.

The computational efficiencies of our adaptive WENO algorithms for the double mach reflection problem (finest grid: 3840×960) are shown in Table 3.4. In this example, we set the coarsest grid to 60×15, the maximum resolution level to $j_{\max} = 6$ and the threshold to $\varepsilon = 5.0 \times 10^{-3}$.

Table 3.4: Computational efficiency of our adaptive algorithm for two dimensional Euler equations.

HRS	WENO-Z
CPU time for finest grid (3840×960) scheme (sec.)	274177.44
CPU time for adaptive algorithm (sec.)	30295.27
Max. percentage of adaptive points over finest points (%)	7.46
CPU gain = CPU time (fine.) / CPU time (adap.)	8.15

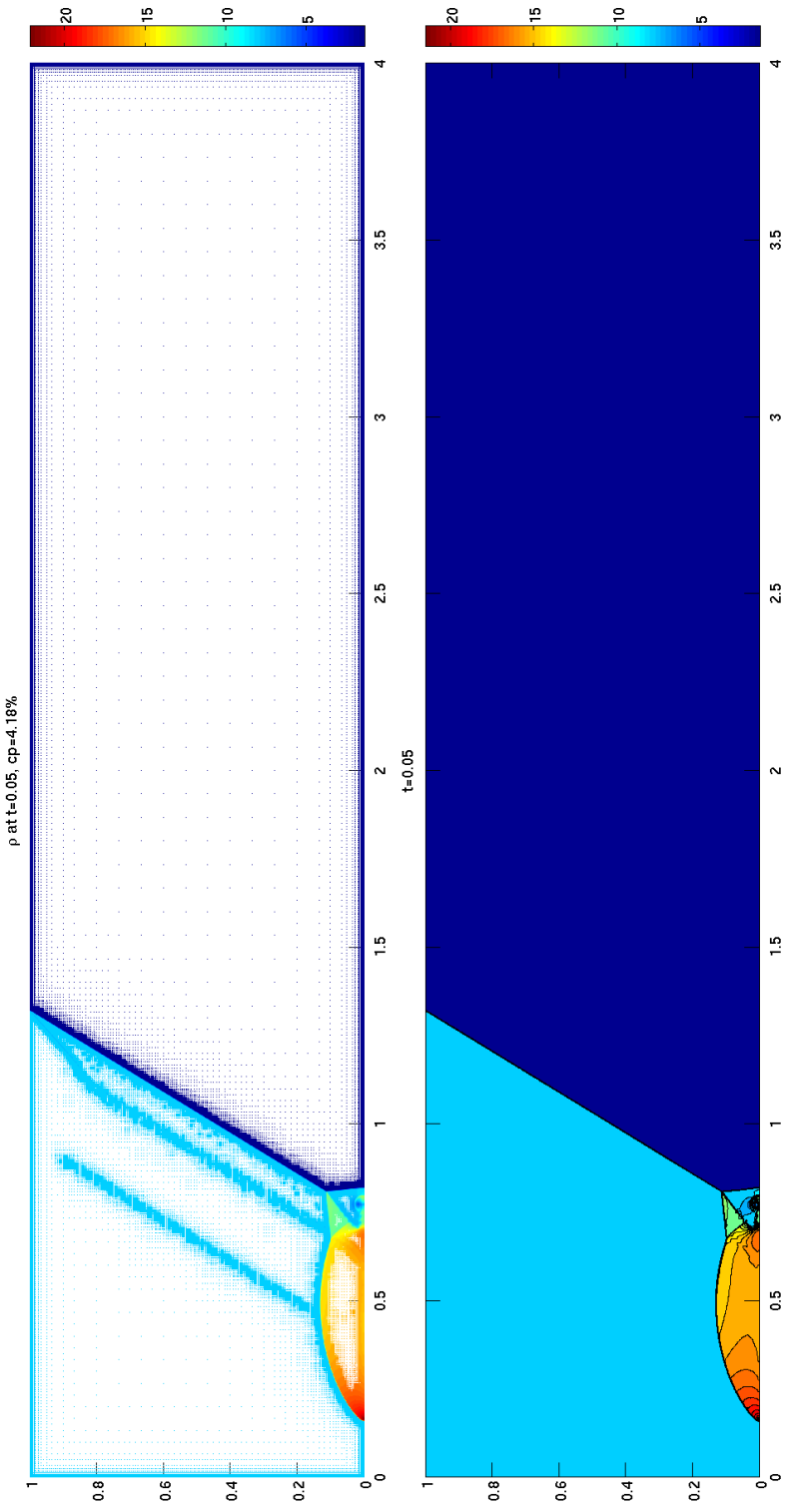


Figure 3.8: Solution for double mach reflection at $t = 0.05$. The density profile obtained with the adaptive WENO-Z algorithm. Top: numerical solutions on adaptive grids; bottom: 30 equally spaced contour lines.

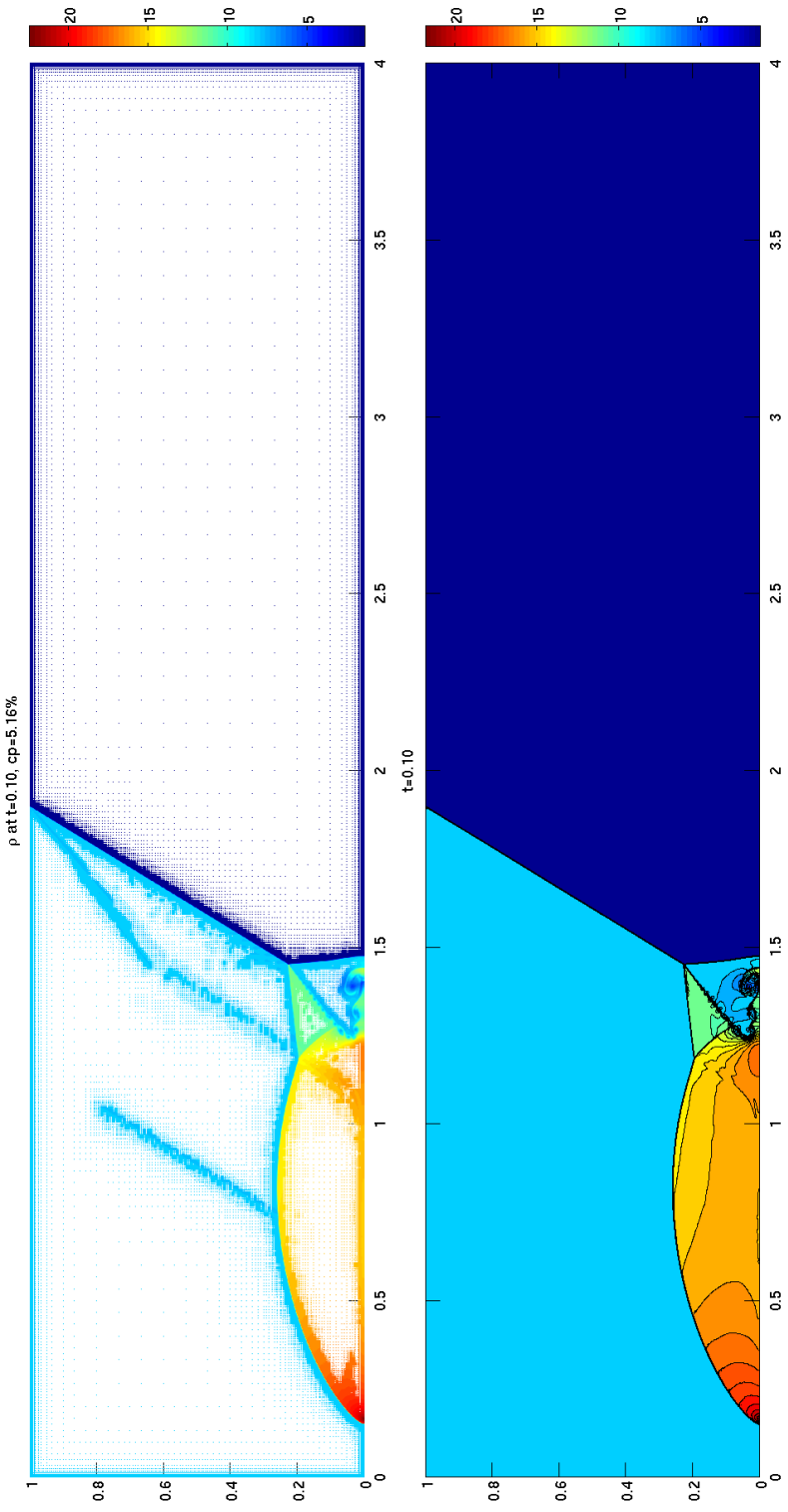


Figure 3.9: Solution for double mach reflection at $t = 0.1$. The density profile obtained with the adaptive WENO-Z algorithm. Top: numerical solutions on adaptive grids; bottom: 30 equally spaced contour lines.

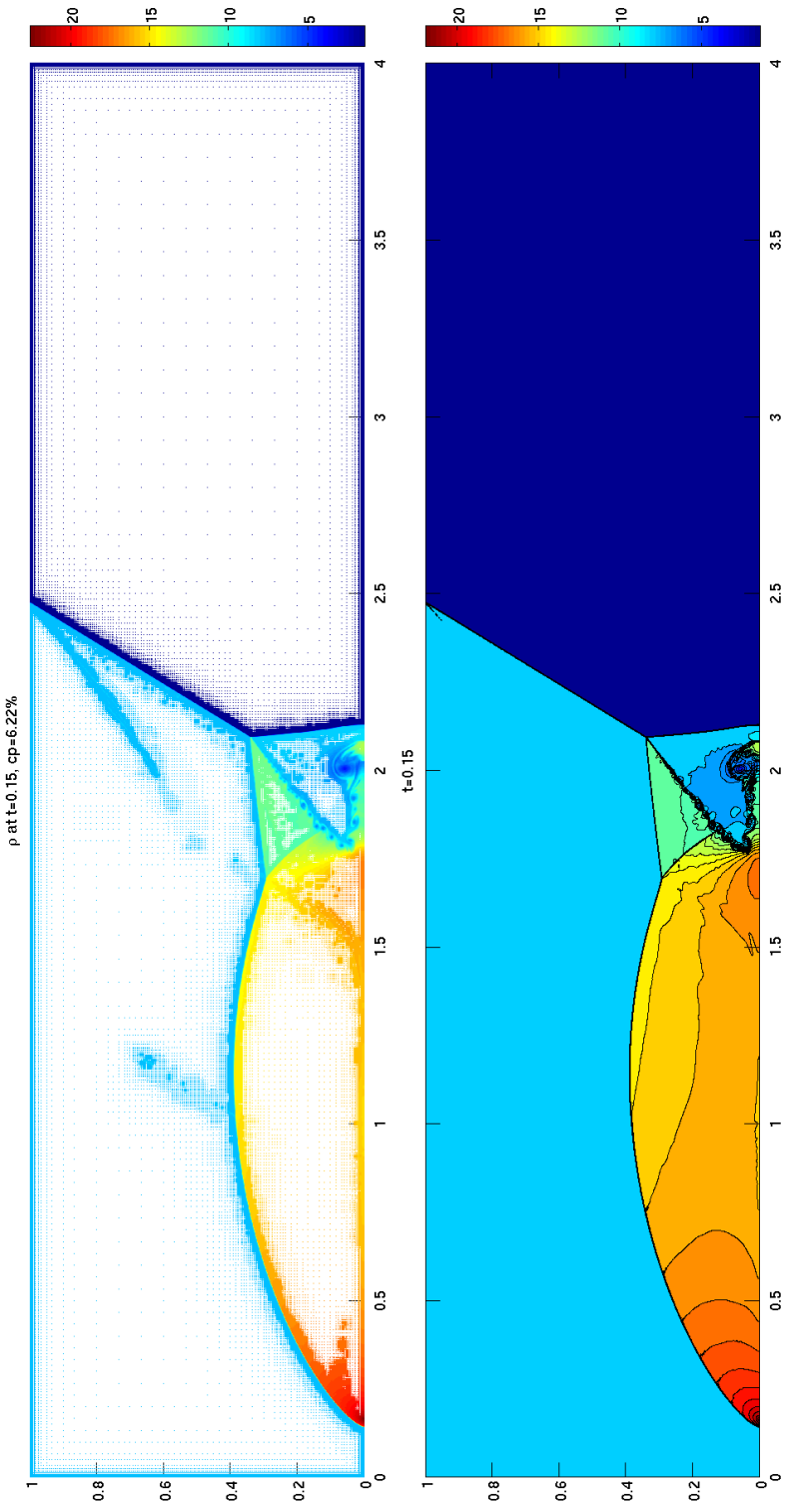


Figure 3.10: Solution for double mach reflection at $t = 0.15$. The density profile obtained with the adaptive WENO-Z algorithm. Top: numerical solutions on adaptive grids; bottom: 30 equally spaced contour lines.

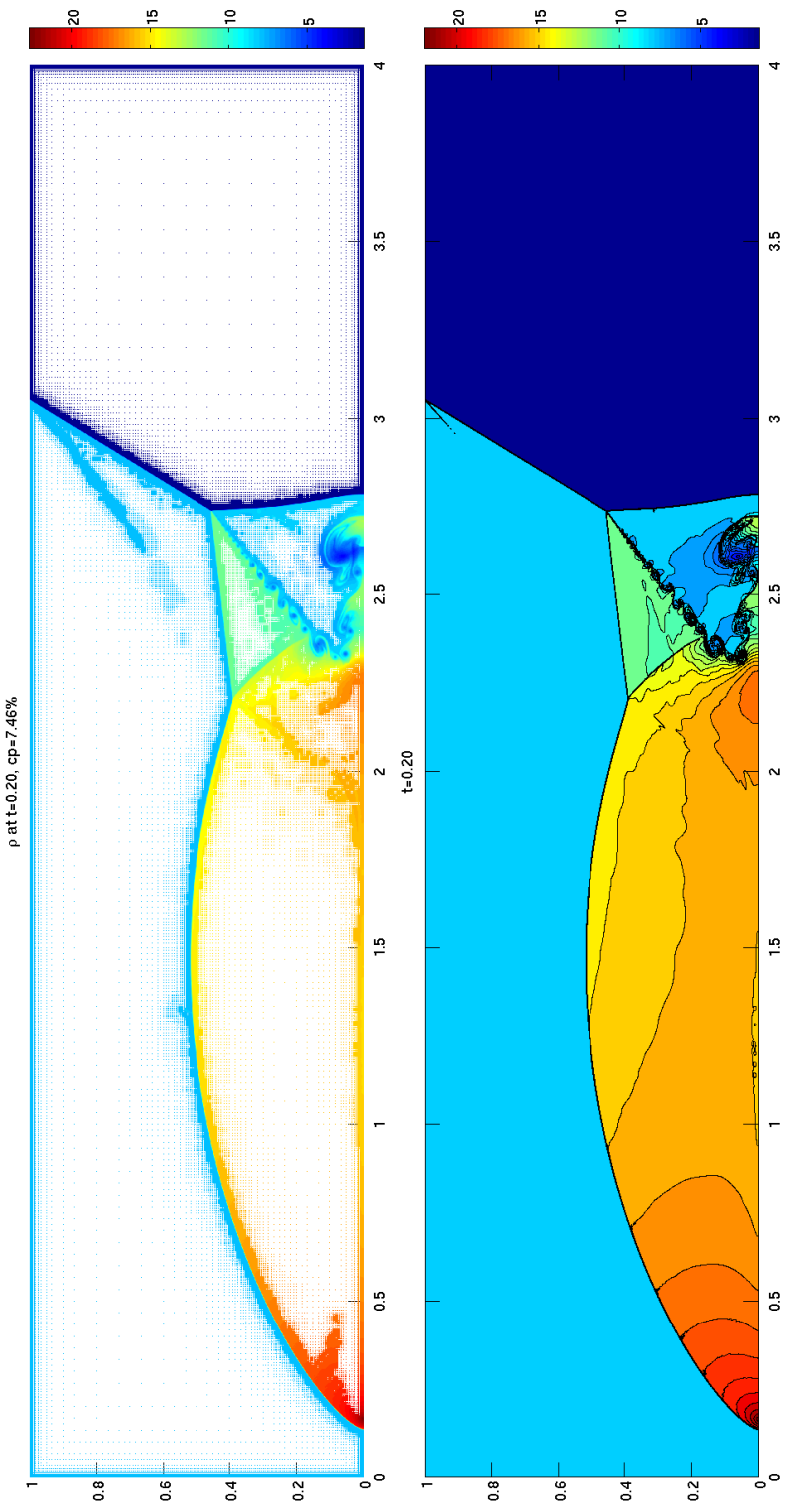


Figure 3.11: Solution for double mach reflection at $t = 0.2$. The density profile obtained with the adaptive WENO-Z algorithm. Top: numerical solutions on adaptive grids; bottom: 30 equally spaced contour lines.

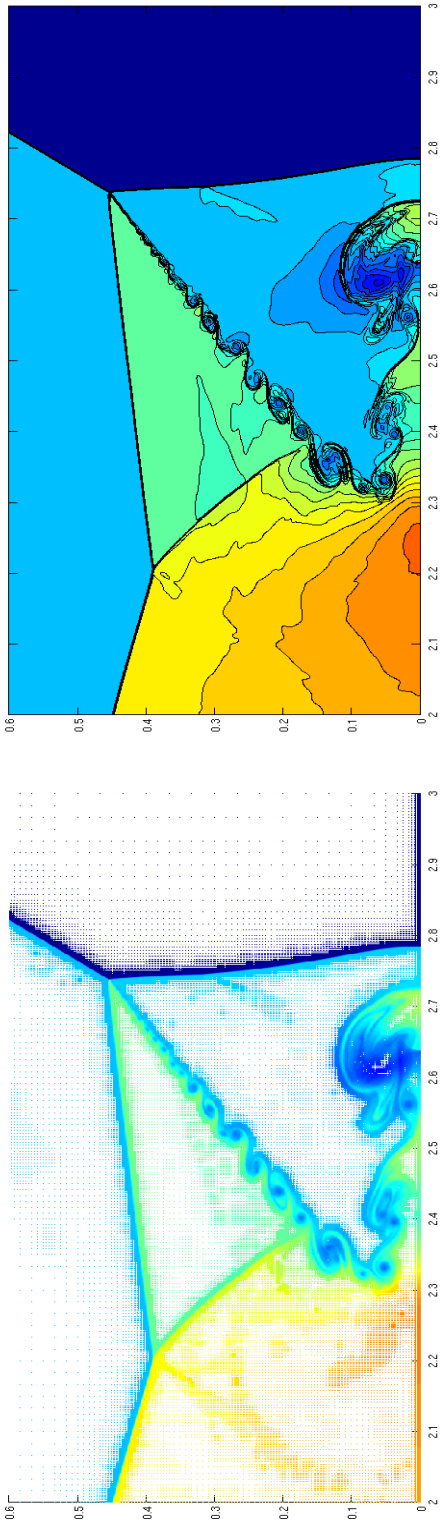


Figure 3.12: Density profile zoomed in. Left: solutions on adaptive grids; right: 30 equally spaced contour lines.

Rayleigh-Taylor instability

Rayleigh-Taylor instability happens on an interface between fluids with different densities when there is an acceleration from heavy fluid to light one [63]. Initial condition is described by :

$$(\rho, u, v, p) = \begin{cases} (2,\ 0,\ -0.025c\cos(8\pi x),\ 2y+1) & y \leq 0.5 \\ (1,\ 0,\ -0.025c\cos(8\pi x),\ y+1.5) & y > 0.5 \end{cases}$$

with a specific heat ratio $\gamma = 5/3$, and c is a speed of sound. Small perturbation is introduced at the y-directional velocity v and the gravitational forcing terms $(0, 0, \rho, \rho v)$ are added to the right hand side of Euler equation (2.2.1). At the top and bottom boundaries, an inflow boundary condition that fixes the physical state is set up and the reflective wall condition is imposed at the left, right boundary. We evolve the conserved quantities until $t = 1.95$ with adaptive parameters chosen as $j_{\max} = 4$, $\epsilon = 5 \times 10^{-3}$ on the coarsest 30×120 grid.

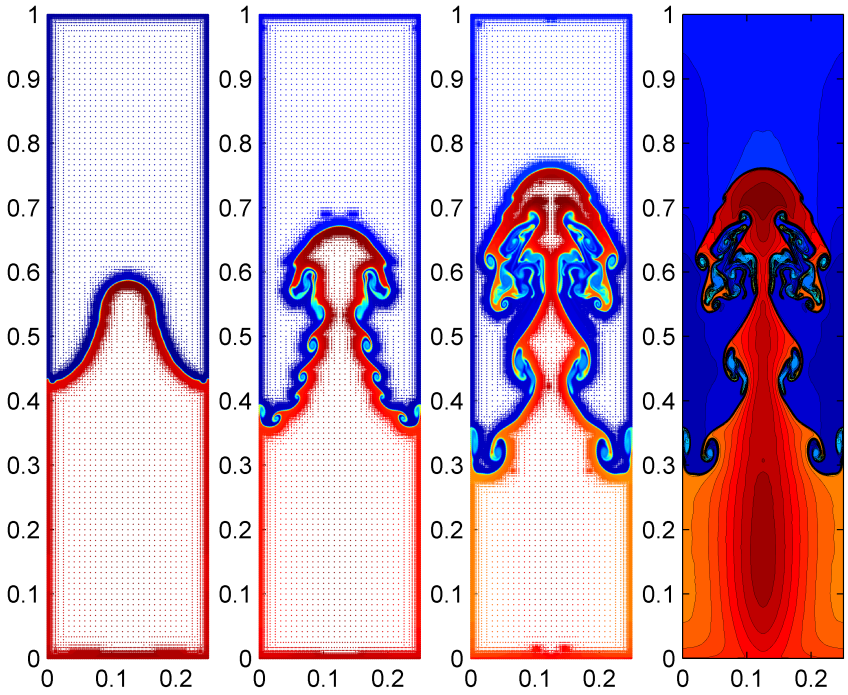


Figure 3.13: Numerical results for Rayleigh-Taylor instability at $t = 0.975, 1.4625, 1.95$.

3.5.3 2-dimensional MHD equations

In order to solve MHD equation, an additional process for divergence-free magnetic field is necessary. From the MHD equations, it is clear that the magnetic field at any time step must have zero divergence, provided the initial magnetic field is divergence-free. However, it is not satisfied automatically from the numerical point of view, especially when artificial oscillations arise and they cause numerical instability near discontinuous region. Thus suitable process for handling divergence-free constraint should be considered.

In our simulations, the divergence cleaning technique [37] which uses an auxiliary scalar field ψ (not an wavelet function) and a partial differential equation to propagate, diffuse divergence error is used. Hence, the MHD

equation is modified as follows :

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho\mathbf{u}\\ E\\ \mathbf{B}\\ \psi\end{pmatrix}+\nabla\cdot\begin{pmatrix}\rho\mathbf{u}\\ \rho\mathbf{u}\mathbf{u}+p^{*}I-\mathbf{B}\mathbf{B}\\ (E+p^{*})\mathbf{u}-(\mathbf{B}\cdot\mathbf{u})\mathbf{u}\\ \mathbf{u}\mathbf{B}-\mathbf{B}\mathbf{u}+\psi I\\ c_h^2\nabla\cdot\mathbf{B}\end{pmatrix}=\begin{pmatrix}0\\ \mathbf{0}\\ 0\\ \mathbf{0}\\ -\dfrac{c_h^2}{c_p^2}\psi\end{pmatrix}.$$

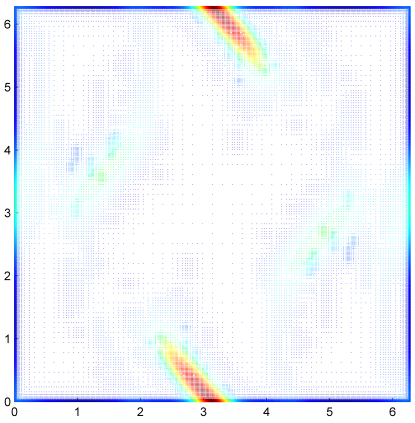
The parameter c_h and c_p are related to the speed of propagation(hyperbolic correction) and diffusion(parabolic correction). We choose parameters $c_h = c_{cfl}\dfrac{\Delta x}{\Delta t}$, and c_p is determined so that $e^{-\Delta t c_h^2/c_p^2} = 0.5$. Also, the last equation is solved in the operator splitting manner.

Orszag-Tang vortex problem

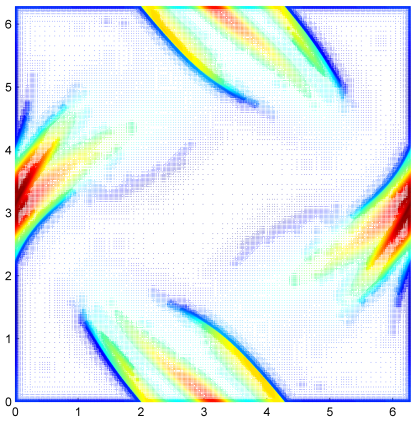
The first MHD test problem is the classical vortex system of Orszag-Tang proposed by [64] and later studied in many papers [65, 66, 67, 64, 37]. The Orszag-Tang vortex problem starts from smooth initial data, but gradually the flow becomes very complicated so that several shock waves is developed. Initial condition on the computational domain $[0, 2\pi] \times [0, 2\pi]$ is described by

$$(\rho, u, v, w, p, B_1, B_2, B_3) = (\gamma^2, -\sin y, \sin x, 0, \gamma, -\sin y, \sin(2x), 0),$$

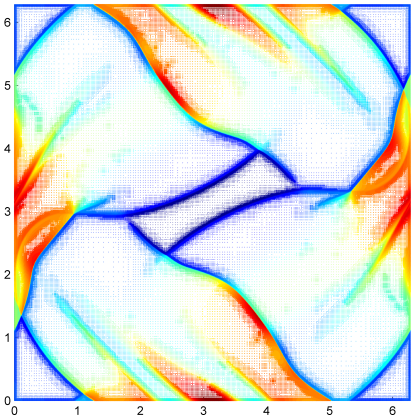
where $\gamma = 5/3$ and final time $= 3$. Periodic boundary condition is imposed at each computational boundary. The results of numerical simulation is achieved with parameter chosen as $j_{\max} = 5$, $\epsilon = 5 \times 10^{-4}$ on the coarsest 40×40 grid. At early stage, an excellent compression is demonstrated. However, the complex vortex structure generating various waves requires more grid points to resolve it while the simulation is being evolved.



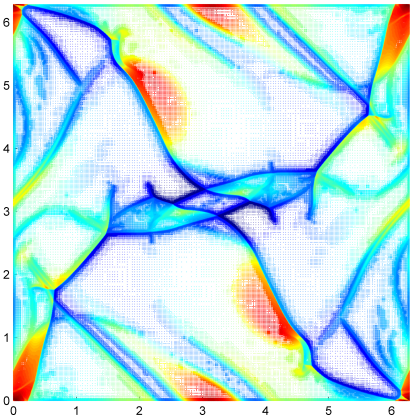
(a) 3.28% used



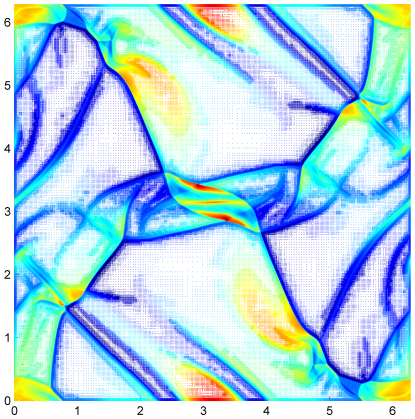
(b) 8.46% used



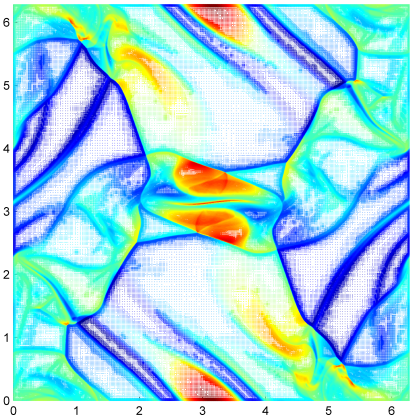
(c) 15.64% used



(d) 23.07% used



(e) 28.08% used



(f) 35.86% used

Figure 3.14: Scattered images displaying only the grid points used to solve OT problem at $t = 0.6, 1.2, 1.8, 2.4, 2.7, 3.0$.

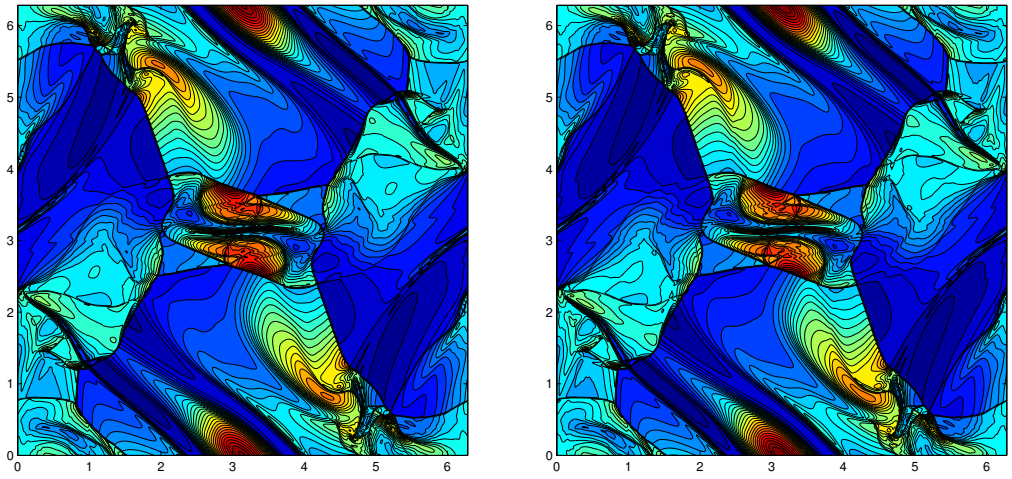


Figure 3.15: The density profiles at $t = 3.0$ for OT problem. In order to plot the left figure, wavelet-based interpolation is performed to Figure 3.14. The right one is computed by full grid WENO method with the 1280×1280 resolution. Good agreement between proposed method and original one can be observed.

Rotor problem

Our last test problem consists of simulating the propagation of strong Alfvén waves rotating near the origins. The problem suggested by [68] involves a dense rapidly spinning cylinder(called a rotor) in an ambient fluid which is stationary and has a light density.

The computational domain is $[0,1] \times [0,1]$ and initial conditions are given by

$$(\rho, u, v) = \begin{cases} \left(10, -\dfrac{v_0}{r_0}(y-0.5), \dfrac{v_0}{r_0}(x-0.5)\right) & r < r_0, \\ (1,0,0) & r > r_1, \\ \left(1+9f(r), -f(r)\dfrac{v_0}{r}(y-0.5), f(r)\dfrac{v_0}{r}(x-0.5)\right) & \text{otherwise}, \end{cases}$$

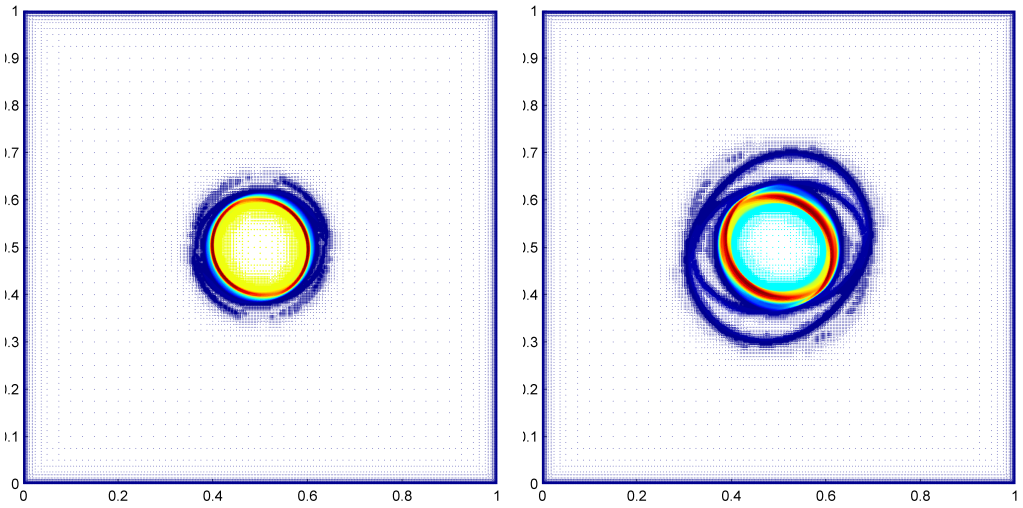
$$(w, p, B_1, B_2, B_3) = (0, 0.5, \frac{2.5}{\sqrt{4\pi}}, 0, 0),$$

where $\gamma = 5/3$, final time $= 0.295$, $v_0 = 1$, $r = \sqrt{(x-0.5)^2 + (y-0.5)^2}$. Parameters r_0, r_1 are used to define the inside, outside regions of the rotor

and set to be $r_0 = 0.1$, $r_1 = 0.115$. Furthermore, $f(r) = \frac{r_1 - r}{r_1 - r_0}$ is employed in order to interpolate the physical values in the transition region of the rotor. At each computational boundary, the outflow boundary condition is set. Figure 3.16 shows the numerical results with adaptation parameters $j_{\max} = 5$, $\epsilon = 5 \times 10^{-4}$ on the coarsest 40×40 grid. Furthermore, the numerical solutions computed by adaptive grid and full grid are displayed in Figure 3.17 to compare the quality of results. One can observe that both solutions agree very well with each other. In the rotor problem, the benefit in the view of efficiency is obvious as displayed in the Figure 3.16, Table 3.5.

Table 3.5: Memory and CPU time comparison of fifth order WENO with adaptive grid and full grid simulations for the rotor problem.

	Adaptive grid	Full grid	Ratio
The maximum number of grid points used	320,214	1,638,400	5.1177
Total CPU time	0.2134	1.0	4.6864



(a) 6.40% of grid points are used

(b) 10.07% of grid points are used

(c) 13.20% of grid points are used

(d) 19.54% of grid points are used

Figure 3.16: Scattered images of the grid points used to solve the rotor problem at $t = 0.0295, 0.0885, 0.1475, 0.295$.

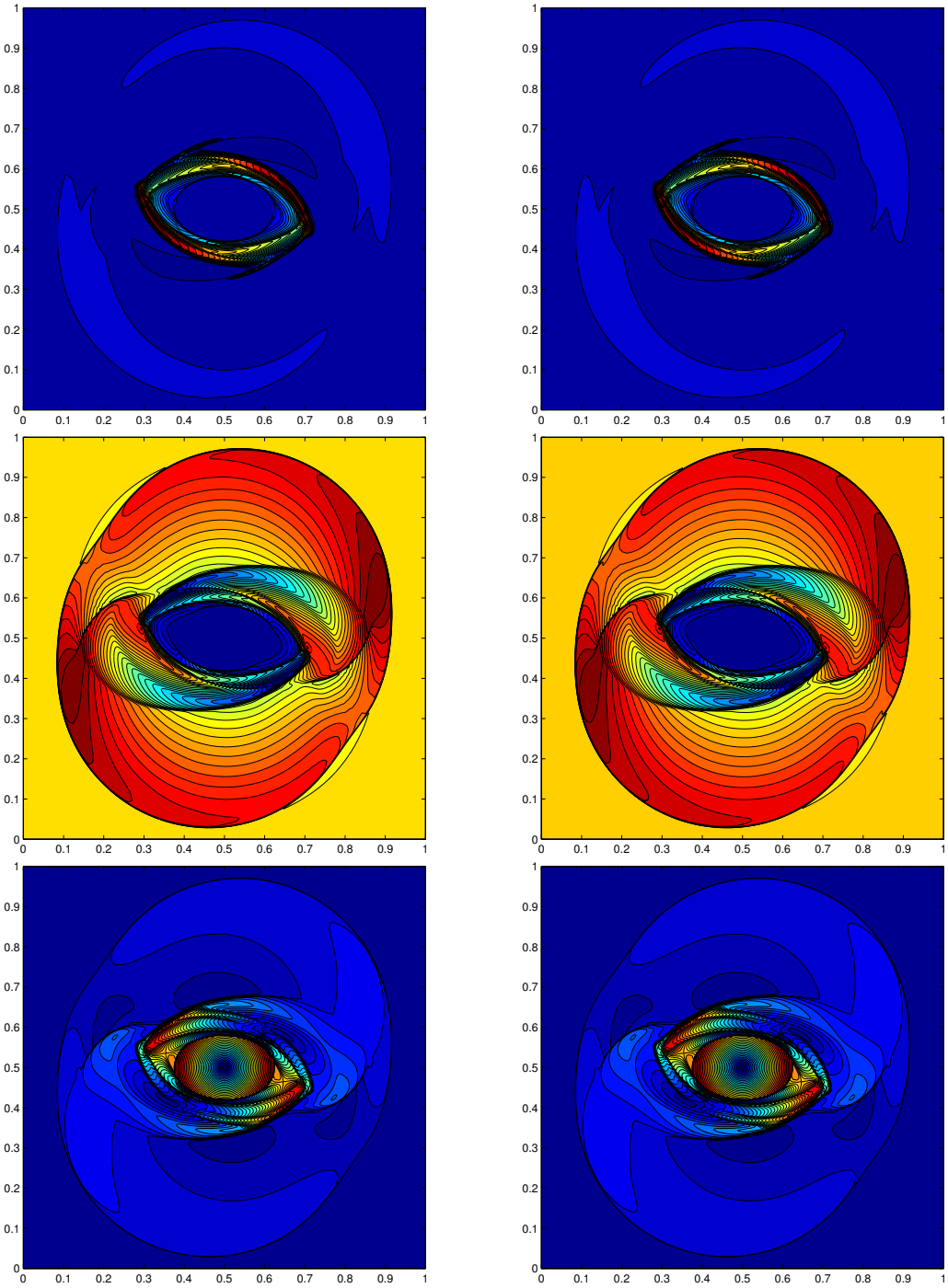


Figure 3.17: Interpolated distributions of density, thermal pressure and mach number for the MHD rotor problem at $t = 0.295$. A comparison of solutions obtained by applying fifth order WENO on adaptive grid(left) and on full grid(right) with the 1280×1280 resolution.

Chapter 4

Combination of Central-Upwind Method and Multi-dimensional Limiting Process

The most popular numerical methods for hyperbolic systems of conservation laws are the upwind scheme originated by Godunov [1] and its extended second-order accurate scheme by van Leer [2]. Another method for the Godunov-type upwind schemes is the Lax-Friedrichs (LxF) central difference scheme introduced by Lax and Fridrichs in [11, 12], in which no Riemann solvers and characteristic decomposition are involved. It has the advantage of simplicity compared with the first-order upwind scheme of Godunov [1]. Unfortunately, the LxF scheme, however, yields large numerical dissipation, which leads to a poor resolution of shock discontinuities and rarefaction waves. In order to reduce the numerical dissipation, Nessyahu and Tadmor proposed a second-order extension of the LxF scheme (refered to NT scheme) in [13], which is based on the staggered form of the Lax-Friedrichs scheme. The NT scheme replaces the first-order piecewise constant solution by van Leer's MUSCL-type piecewise-linear one to construct second-order approximation avoiding oscillations at discontinuities and achieving a

sharp and accurate shock capturing. The NT scheme retains the simplicity of Riemann-free LxF framework and avoids the disadvantage of excessive first order dissipation of the LxF scheme. In the year 2000, Kurganov and Tadmor (CU-KT) [14] proposed modifications of NT scheme which has a smaller amount of numerical viscosity than that of the original NT scheme. The second-order KT schemes with a semidiscrete formulation were based on integration over Riemann fans of variable sizes and used more precise information for the local speeds of propagation. Extensions to multidimensional problems were introduced in [15].

Later generalization of the CU-KT schemes proposed by Kurganov, Noelle and Petrova [16] utilizes one-sided local speeds of propagation(referred to CU-KNP scheme). The KT scheme uses only the maximum wave speed for computing numerical fluxes while the CU-KNP scheme uses both maximal and minimum wave speeds to reduce the numerical viscosity. The schemes presented by Liu and Osher [17] are higher than second order based on the non-oscillatory third-order reconstruction with staggered evolution of the reconstructed cell averages[18]. High-order essentially non-oscillatory (ENO) [19] and weighted ENO (WENO) [3],[20] reconstructions were combined with central schemes by Bianco, Puppo and Russo [21]. Modifications of central schemes of Bianco, Puppo and Russo [21] were introduced in [22] which is a central weighted non-oscillatory (CWENO) reconstruction. Other central schemes with WENO are presented in [23, 24] and extensions to multidimensional problems can be found in [25, 26]. The higher order reconstructions of central schemes result in decreasing numerical dissipations and yield a higher resolution of shocks, rarefactions, and other spontaneous evolution of large gradient phenomena almost as sharp as other higher order upwind schemes.

Most of the numerical schemes for hyperbolic conservation laws have been developed on one-dimensional flow physics, so that two-dimensional flows are usually extended straightforward by dimensional splitting techniques. Although this approach allows the rigorous analysis of numerical schemes, one-dimensional limiters may fail to achieve a good shock resolution if the shock is located in direction diagonal to the computational grid.

These numerical schemes lead to insufficient or excessive numerical dissipation due to the essential limitations of the accurate and efficient calculations for multi-dimensional flows. During the last two decades there were activities to control numerical oscillations and to design multi-dimensional limiter functions [27, 28, 29]. As successive studies, Kim and Kim extended the one-dimensional monotonicity condition to two-dimensional flow and proposed multi-dimensional limiting process (MLP) for the two-dimensional compressible Euler [30]. They also proposed a multi-dimensional limiting process (MLP) in the same article which used multi-dimensional informations for slope limitation to control the oscillations across discontinuities for multi-dimensional applications. Furthermore, improved MLP limiters have been devised [31, 32, 33, 34] be efficiently implemented in three-dimensional space. The main focus of the MLP methods is to eliminate excessive numerical dissipations and upgrade solution accuracy by predicting the physical distributions of flow variables in multi-space dimensions.

In this chapter, we study multi-dimensional limiting process introduced by Yoon et al. [34] and derive a central scheme which is adopts the modification of the multi-dimensional limiting process (MLP). Since the CU-KNP scheme has a relatively computational cheap cost and a high numerical stability, the combination of both approaches promises a good monotone shock capturing and a good convergence behavior at relatively cheap computational cost. Some improvements over standard CU-KNP [16] will be given and a corresponding modified CU-KNP scheme (compared to CU-MLP) will be presented. We see that this modification yields better results than the original CU-KNP schemes. Some numerical experiments are conducted to demonstrate the performance of the proposed scheme.

4.1 Review of Central-Upwind method

In this section, we briefly review the semi-discrete central-upwind scheme for two-dimensional compressible flow suggested by Kurganov at el. [16]. We

consider hyperbolic conservation laws in two-dimensional space

$$u_t + f(u)_x + g(u)_y = 0. \tag{4.1.1}$$

Central-upwind schemes can be easily implemented only with a knowledge of the physical flux functions and the extremal eigenvalues of the flux Jacobian associated with a hyperbolic system of conservation laws. We discretize using a Cartesian grid with uniform spacing $\Delta x(= x_i - x_{i-1})$ and $\Delta y(= y_j - y_{j-1})$, and the intermediate points $x_{i\pm\frac{1}{2}} = x_i \pm \frac{\Delta x}{2}$ and $y_{j\pm\frac{1}{2}} = y_j \pm \frac{\Delta y}{2}$. We also define the cell averages based on spatial cells $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$,

$$\overline{u}_{ij}^n = \frac{1}{\Delta x \Delta y} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} u(x, y, t^n) dy dx. \tag{4.1.2}$$

To advance the computation to the next time level $t = t^{n+1}$, we reconstruct a piecewise linear polynomial $\widetilde{u}(x, y, t^n)$ of the form

$$\widetilde{u}(x, y, t^n) = \sum_{i,j} \left[\overline{u}_{ij}^n + (u_x)_{ij}^n (x - x_i) + (u_y)_{ij}^n (y - y_j) \right] \chi_{ij}(x, y) \tag{4.1.3}$$

where

$$\begin{aligned}
(u_x)_{ij}^n &= \frac{1}{\Delta x} \mathbf{minmod} \left(\alpha \Delta \overline{u}_{i+1/2,j}, \frac{1}{2}(\Delta \overline{u}_{i+1/2,j} + \Delta \overline{u}_{i-1/2,j}), \alpha \Delta \overline{u}_{i-1/2,j} \right), \\
(u_y)_{ij}^n &= \frac{1}{\Delta y} \mathbf{minmod} \left(\alpha \Delta \overline{u}_{i,j+1/2}, \frac{1}{2}(\Delta \overline{u}_{i,j+1/2} + \Delta \overline{u}_{i,j-1/2}), \alpha \Delta \overline{u}_{i,j-1/2} \right),
\end{aligned}$$

with $\alpha \in [1, 2]$ and $\Delta \overline{u}_{i+1/2,j} = \overline{u}_{i+1,j} - \overline{u}_{ij}$. Here, $\chi_{ij}(x, y)$ denotes the characteristic functions of the corresponding interval $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$ and the multivariable **minmod** function is defined by

$$\mathbf{minmod}(x_1, x_2, \cdots) = \begin{cases} \min\{x_i\} & \text{if } x_i > 0 \text{ for all } i, \\ \max\{x_i\} & \text{if } x_i < 0 \text{ for all } i, \\ 0 & \text{otherwise.} \end{cases}$$

The piecewise linear interpolant $\widetilde{u}$ may have discontinuities along the line

$x = x_{i\pm\frac{1}{2}}$ and $y = y_{j\pm\frac{1}{2}}$ which propagate with different right and left going local wave speeds. To compute the corresponding local wave speeds, one may use

$$\begin{aligned}
a^+_{i+\frac{1}{2},j} &= \max\left[\lambda^x_N\left(\frac{\partial f}{\partial u}(u^L_{i+\frac{1}{2},j})\right), \lambda^x_N\left(\frac{\partial f}{\partial u}(u^R_{i+\frac{1}{2},j})\right), 0\right],\\
a^-_{i+\frac{1}{2},j} &= \min\left[\lambda^x_1\left(\frac{\partial f}{\partial u}(u^L_{i+\frac{1}{2},j})\right), \lambda^x_1\left(\frac{\partial f}{\partial u}(u^R_{i+\frac{1}{2},j})\right), 0\right],\\
b^+_{i,j+\frac{1}{2}} &= \max\left[\lambda^y_N\left(\frac{\partial g}{\partial u}(u^L_{i,j+\frac{1}{2}})\right), \lambda^y_N\left(\frac{\partial g}{\partial u}(u^R_{i,j+\frac{1}{2}})\right), 0\right],\\
b^-_{i,j+\frac{1}{2}} &= \min\left[\lambda^y_1\left(\frac{\partial g}{\partial u}(u^L_{i,j+\frac{1}{2}})\right), \lambda^y_1\left(\frac{\partial g}{\partial u}(u^R_{i,j+\frac{1}{2}})\right), 0\right],
\end{aligned} \tag{4.1.4}$$

where $\lambda^x_N(\lambda^y_N)$ and $\lambda^x_1(\lambda^y_1)$ denote the largest and the smallest eigenvalues of the Jacobian $\frac{\partial f}{\partial u}(\frac{\partial g}{\partial u})$, respectively. The point values of the piecewise linear reconstruction (4.1.1) are given by

$$\begin{aligned}
u^L_{i+\frac{1}{2},j} &= \overline{u}_{ij} + 0.5(u_x)^n_{ij}\Delta x,\\
u^R_{i+\frac{1}{2},j} &= \overline{u}_{i+1,j} - 0.5(u_x)^n_{i+1,j}\Delta x,\\
u^L_{i,j+\frac{1}{2}} &= \overline{u}_{ij} + 0.5(u_y)^n_{ij}\Delta y,\\
u^R_{i,j+\frac{1}{2}} &= \overline{u}_{i,j+1} - 0.5(u_y)^n_{i,j+1}\Delta y.
\end{aligned} \tag{4.1.5}$$

The two-dimensional semi-discrete schemes can be written in the following conservation form:

$$\frac{d}{dt}u_{ij} \;=\; -\frac{\widehat{f}_{i+\frac{1}{2},j}(t) - \widehat{f}_{i-\frac{1}{2},j}(t)}{\Delta x} - \frac{\widehat{g}_{i,j+\frac{1}{2}}(t) - \widehat{g}_{i,j-\frac{1}{2}}(t)}{\Delta y} \tag{4.1.6}$$

where the numerical flux functions are

$$\begin{aligned}
\widehat{f}_{i+\frac{1}{2},j}(t) &= \frac{a^+_{i+\frac{1}{2},j}f(u^L_{i+\frac{1}{2},j}) - a^-_{i+\frac{1}{2},j}f(u^R_{i+\frac{1}{2},j})}{a^+_{i+\frac{1}{2},j} - a^-_{i+\frac{1}{2},j}} + \frac{a^+_{i+\frac{1}{2},j}a^-_{i+\frac{1}{2},j}}{a^+_{i+\frac{1}{2},j} - a^-_{i+\frac{1}{2},j}}\left[u^R_{i+\frac{1}{2},j} - u^L_{i+\frac{1}{2},j}\right],\\
\widehat{g}_{i,j+\frac{1}{2}}(t) &= \frac{b^+_{i,j+\frac{1}{2}}g(u^L_{i,j+\frac{1}{2}}) - b^-_{i,j+\frac{1}{2}}g(u^R_{i,j+\frac{1}{2}})}{b^+_{i,j+\frac{1}{2}} - b^-_{i,j+\frac{1}{2}}} + \frac{b^+_{i,j+\frac{1}{2}}b^-_{i,j+\frac{1}{2}}}{b^+_{i,j+\frac{1}{2}} - b^-_{i,j+\frac{1}{2}}}\left[u^R_{i,j+\frac{1}{2}} - u^L_{i,j+\frac{1}{2}}\right].
\end{aligned} \tag{4.1.7}$$

We have presented above CU-KNP semi-discrete approximation of the spatial discretization for systems of conservation laws. For the time discretization, we can employ several methods for solving an Ordinary Differential Equation(ODE):

$$u_t = \mathcal{L}(u, t).$$

Here we have used a 3rd-order TVD Runge-Kutta type discretization for time stepping to solve the ODE [4]:

$$\begin{aligned} u^{(1)} &= u^n + \Delta t \mathcal{L}(u^n), \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t \mathcal{L}(u^{(1)}), \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t \mathcal{L}(u^{(2)}). \end{aligned} \tag{4.1.8}$$

4.2 Review of Multi-dimensional Limiting Process

In this section, we describe the basic concept of MLP limiter in two-dimensional space [34]. Generally, most of the numerical methods for two or three dimensional hyperbolic conservation laws are considered by dimensional splitting ways. This approach is an efficient computational method but has clear disadvantages. The scheme is insufficient to control oscillations near shock discontinuity and maybe fail to achieve a good shock resolution if the shock is located in direction diagonal to the computational grid in multi-dimensional space. When TVD concept is extended to multi-dimensional flow by dimensional splitting manner, it cannot guarantee monotonic solutions. In this aspect, there should be a consideration to control the oscillation in multi-dimensional space situation. To overcome this drawback, the multi-dimensional limiting process (MLP) was originally proposed by Kim et al. [30] and Yoon et al. [34] to reinforce oscillation control in multi-dimensional flows. MLP is similar to a conventional second order limiters such as superbee or minmod but it uses diagonal volume informations. Before introducing the concept of MLP, we consider the cell-interface value u with symmetric MUSCL type TVD limiter

:

$$\begin{aligned} u^L_{i+1/2,j} = &\quad \overline{u}_{ij} + 0.5\phi(r^L_x)\Delta\overline{u}_{i-1/2,j}, \\ u^R_{i+1/2,j} = &\ \overline{u}_{i+1,j} - 0.5\phi(r^R_x)\Delta\overline{u}_{i+3/2,j}, \end{aligned} \tag{4.2.1}$$

with $\Delta\overline{u}_{i-1/2,j} = \overline{u}_{ij} - \overline{u}_{i-1,j}$, $r^L_x = \dfrac{\Delta\overline{u}_{i+1/2,j}}{\Delta\overline{u}_{i-1/2,j}}$ and $r^R_x = \dfrac{\Delta\overline{u}_{i+1/2,j}}{\Delta\overline{u}_{i+3/2,j}}$. Here $\phi(r)$ is a limiting function and satisfies the symmetric condition $\phi(r) = r\phi(1/r)$ so that we can obtain

$$u^R_{i-1/2,j} = \overline{u}_{ij} - 0.5\phi(r^L_x)\Delta\overline{u}_{i-1/2,j}.$$

The cell interface values of equation (4.2.1) may be interpreted as a first order upwind representation plus an additional anti-diffusive term.

The Sweby's TVD zone [69] of one-dimensional limiting condition is given by

$$0 \le \phi(r) \le \min(2r, 2). \tag{4.2.2}$$

Since the extension of (4.2.2) in a dimensional splitting manner is insufficient to prevent oscillations in multi-dimensional flow, it should be modified and extended with appropriate consideration of multidimensional situation. To obtain monotonic solution in multi-dimensional space, MLP should satisfy the discrete maximum principle

$$\overline{u}^{\min}_{nbd} \le u \le \overline{u}^{\max}_{nbd} \tag{4.2.3}$$

where u is the estimated value at a vertex point. Here $\overline{u}^{\min}_{nbd}$ and $\overline{u}^{\max}_{nbd}$ are the minimum and maximum values, respectively, among the neighboring cell-averaged values sharing the vertex. The vertex point values are then expressed in terms of cell-averaged values and variations within a cell. A physical property at each vertex is then estimated by summing the monotonic variations along each coordinate direction, and the MLP condition (4.2.3) is applied. To obtain TVD regions of the multi-dimensional space, a detailed

analysis is performed through the one dimensional limiter functions. We have

$$0 \leq \phi(r) \leq \min(\alpha r, \alpha). \tag{4.2.4}$$

α is the multi-dimensional restriction coefficient which determines the baseline variable limiting region. From (4.2.2) and (4.2.4), one can see that the MLP limiting region is determined by depending on local multi-dimensional flow physics, while TVD provides a fixed limiting region. For practical reasons, the range of α is in $[0, 2]$. Due to $\phi(r) = 0$ for $r < 0$ the limiter switches to first order accuracy if r becomes negative. This is exactly the case for any extreme point.

For the third and higher order reconstructions, Kim et al. [30] introduces local slope variable β to filter the unlimited values by the MLP condition (4.2.3) and (4.2.4). They also introduced the MLP limiter function $\phi(r_x)$ in the x-direction as

$$\phi^{L(R)}(r_x) = \max(0, \min(\alpha_x, \alpha_x r_x, \beta_x^{L(R)})). \tag{4.2.5}$$

Finally we summarize the MLP methods proposed by [34] for two-dimensional compressible flow. Cell-interface value of u is obtained as follows.

$$\begin{aligned} u^L_{i+1/2,j} &= \overline{u}_{ij} + 0.5\phi^L\left((r_x^L)_{ij}\right)\Delta\overline{u}_{i-1/2,j}, \\ u^R_{i-1/2,j} &= \overline{u}_{ij} - 0.5\phi^R\left((r_x^R)_{ij}\right)\Delta\overline{u}_{i+1/2,j}, \end{aligned} \tag{4.2.6}$$

where MLP limiter functions $\phi(\cdot)$ are given by

$$\begin{aligned} \phi^L\left((r_x^L)_{ij}\right) &= \max\left(0, \min((\alpha_x)_{ij}, (\alpha_x)_{ij}(r_x^L)_{ij}, (\beta_x^L)_{ij}\right), \\ \phi^R\left((r_x^R)_{ij}\right) &= \max\left(0, \min((\alpha_x)_{ij}, (\alpha_x)_{ij}(r_x^R)_{ij}, (\beta_x^R)_{ij}\right), \end{aligned} \tag{4.2.7}$$

where $(\alpha_x^L)_{ij} = (\alpha_x)_{ij}$, $(\alpha_x^R)_{ij} = (\alpha_x)_{ij}$, $(r_x^L)_{ij} = (r_x)_{ij}$ and $(r_x^R)_{ij} = 1/(r_x)_{ij}$. Here, $(\alpha_x)_{ij}$ is defined by

$$(\alpha_x)_{ij} = \left|\frac{2\max\{1, (r_x)_{ij}\}}{\{1 + (r_{xy})_{ij}\}\Delta\overline{u}_{i+1/2,j}}\right| \min\left[|\overline{u}^{\max}_{\kappa_x,\kappa_y} - \overline{u}_{ij}|, |\overline{u}^{\min}_{\kappa'_x,\kappa'_y} - \overline{u}_{ij}|\right], \tag{4.2.8}$$

Table 4.1: Convergence Order.

Order of accuracy	3	5
$(\beta_x^L)_{ij}$	$\dfrac{1+2(r_x)_{ij}}{3}$	$\dfrac{-2/r_x+11+24(r_x)_{ij}-3(r_x)_{ij}(r_x)_{i+1,j}}{30}$
$(\beta_x^R)_{ij}$	$\dfrac{1+2(r_x^R)_{ij}}{3}$	$\dfrac{-2/r_x^R+11+24(r_x^R)_{ij}-3(r_x^R)_{ij}(r_x^R)_{i+1,j}}{30}$

where $(r_x)_{ij}=\dfrac{\Delta\overline{u}_{i+1/2,j}}{\Delta\overline{u}_{i-1/2,j}}$, $(r_{xy})_{ij}=\dfrac{\Delta\overline{u}^y_{i,j+\kappa_y/2}}{\Delta\overline{u}^x_{i+\kappa_x/2,j}}$ and

$$\begin{aligned}\overline{u}^{\min}_{\kappa_x',\kappa_y'}&=\min(\overline{u}_{ij},\overline{u}_{i+\kappa_x',j},\overline{u}_{i,j+\kappa_y'},\overline{u}_{i+\kappa_x',j+\kappa_y'}),\\ \overline{u}^{\max}_{\kappa_x,\kappa_y}&=\max(\overline{u}_{ij},\overline{u}_{i+\kappa_x,j},\overline{u}_{i,j+\kappa_y},\overline{u}_{i+\kappa_x,j+\kappa_y}).\end{aligned}\tag{4.2.9}$$

Here the cell-average values $\Delta\overline{u}^x_{i+\kappa_x/2,j}$ and $\Delta\overline{u}^y_{i,j+\kappa_y/2}$ are expressed by

$$\begin{aligned}\Delta\overline{u}^x_{i+\kappa_x/2,j}&=u^*_{i+\kappa_x/2,j}-\overline{u}_{ij},\\ \Delta\overline{u}^y_{i,j+\kappa_y/2}&=u^*_{i,j+\kappa_y/2}-\overline{u}_{ij}\end{aligned}\tag{4.2.10}$$

where $u^*_{i+\kappa_x/2,j}$ and $u^*_{i,j+\kappa_y/2}$ are temporary cell-interface values at $(i+\kappa_x/2,j)$ and $(i,j+\kappa_y/2)$ respectively. For the computational efficiency

$$r_{xy}\approx\left|\frac{\overline{u}_{i,j+1}-\overline{u}_{i,j-1}}{\overline{u}_{i+1,j}-\overline{u}_{i-1,j}}\right|$$

is a reasonable choice. Furthermore, $(\beta_x^L)_{ij}$ and $(\beta_x^R)_{i+1,j}$ are determined by the third-order or the fifth-order polynomial interpolation given in TABLE 4.1. Numerical fluxes in y-direction can be obtained similarly.

4.3 Central-Upwind method with Modified MLP limiter

In the MUSCL-type linear reconstruction, local extrema occur at vertex, and thus only the vertex values are limited by the MLP condition. By considering all neighboring cells sharing the vertex, the range of the multi-dimensional

slope limiter can be obtained. From the basic point of the MLP, the limiting condition in multi-dimensional space is given by

$$\overline{u}^{\min}_{\kappa_x,\kappa_y} \le u_{i+\kappa_x/2,j+\kappa_y/2} \le \overline{u}^{\max}_{\kappa_x,\kappa_y}, \tag{4.3.1}$$

where $u_{i+\kappa_x/2,j+\kappa_y/2}$ is a vertex point value. Indeed the limiting condition (4.3.1) is applied to the four vertex points $(i+\kappa_x/2, j+\kappa_y/2), \kappa_x, \kappa_y = \pm 1$. Here $\overline{u}^{\min}_{\kappa_x,\kappa_y}$ and $\overline{u}^{\max}_{\kappa_x,\kappa_y}$ are the minimum and maximum cell-averaged values around the vertex point $(i+\kappa_x/2, j+\kappa_y/2)$. The vertex point value is calculated by the cell-averaged value and variations within a cell:

$$u_{i+\kappa_x/2,j+\kappa_y/2} = \overline{u}_{ij} + \Delta u^x_{i+\kappa_x/2,j} + \Delta u^y_{i,j+\kappa_y/2}. \tag{4.3.2}$$

Also, the variations (4.2.10) combined with MLP condition (4.2.4) can be defined by

$$\begin{aligned} \Delta u^x_{i+\kappa_x/2,j} &= 0.5\kappa_x \phi(r_x)\Delta\overline{u}_{i-1/2,j}, \\ \Delta u^y_{i,j+\kappa_y/2} &= 0.5\kappa_y \phi(r_y)\Delta\overline{u}_{i,j-1/2}. \end{aligned} \tag{4.3.3}$$

Using variations and employing the TVD-MUSCL limiting, we obtain (4.3.4) from (4.3.2),

$$u_{i+\kappa_x/2,j+\kappa_y/2} = \overline{u}_{ij} + 0.5\kappa_x\phi(r_x)\Delta\overline{u}_{i-1/2,j} + 0.5\kappa_y\phi(r_y)\Delta\overline{u}_{i,j-1/2}. \tag{4.3.4}$$

Since the vertex point values of (4.3.4) does not satisfy the limiting condition (4.3.1) at all vertex points, we need to restrict the range of the limiter function ϕ. For the limiter function in (4.3.4) and (4.2.5), the parameters α_x and α_y are usually taken by 2, respectively, so that it satisfies one-dimensional TVD property. However, those choices of α_x, α_y do not guarantee maximum property of two-dimensional space (see Fig.4.1).

As a result, to control $\phi(r_x)$ and $\phi(r_y)$ for satisfying the multi-dimensional limiting condition of MLP we have to determine the parameters α_x and α_y condignly. In [34], Yoon et al. introduced single values of parameters α_x and α_y. Later, Gerlinger suggested the different parameter values for both

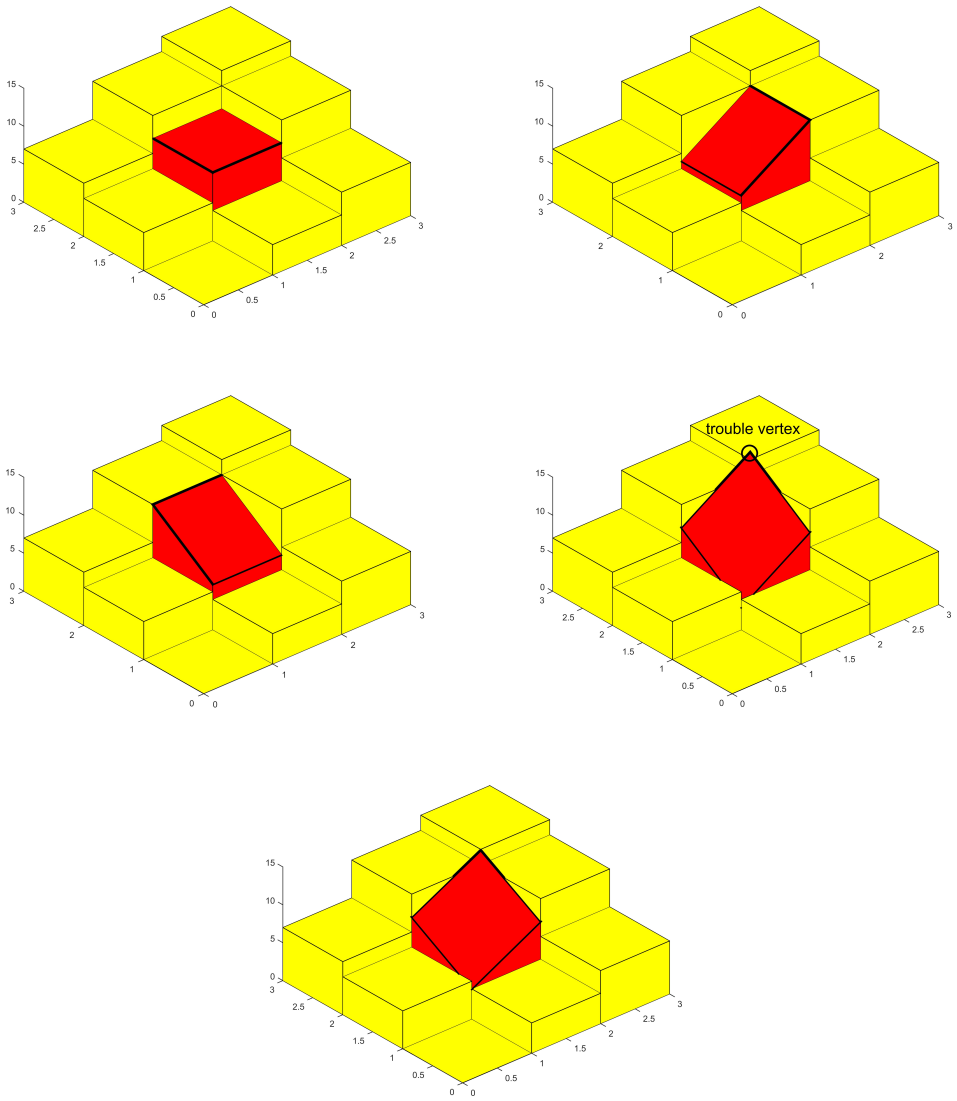


Figure 4.1: Main concept of the MLP : (1)Initial profile, (2)TVD limiting along x-direction, (3)TVD limiting along y-direction, (4)TVD limiting along xy-direction and (5)Adjusting α_x and α_y. In general, applying 1D TVD limiting in the dimensional splitting manner does not guarantee TVD property along the diagonal direction.

coordinate directions in [31] to satisfy the maximum property (4.3.1), to avoid local extrema at the corner points of a volume and to prevent excessive numerical dissipation. Those parameters α_x and α_y, however, are smaller

than those of MLP condition. So, the numerical results are more dissipative than necessary in near discontinuities. In contrast to previous parameters, the maximizations of α_x and α_y are introduced here for reducing the dissipation near discontinuities by modifying the limiting condition.

For the variable limiting regions, it is necessary to find appropriate values of the parameters α_x and α_y in the limiter function to prevent overestimating feature that disturb TVD. We list several properties to lessen computational effort for choosing α_x and α_y satisfying MLP condition.

Proposition 4.3.1. *Whenever* $r_{xy} < 0$ *in* (4.2.8)*, the limiting condition* (4.3.1) *holds automatically.*

Proof. Without loss of generality, we may assume that

$$\Delta u^x_{i+\kappa_x/2,j} > 0 \text{ and } \Delta u^y_{i,j+\kappa_y/2} \leq 0.$$

Then we have

$$\begin{aligned}\overline{u}_{ij} + \Delta u^y_{i,j+\kappa_y/2} &\leq u_{i+\kappa_x/2,j+\kappa_y/2} \\ &= \overline{u}_{ij} + \Delta u^x_{i+\kappa_x/2,j} + \Delta u^y_{i,j+\kappa_y/2} \\ &\leq \overline{u}_{ij} + \Delta u^x_{i+\kappa_x/2,j}.\end{aligned}$$

Since the variations $\Delta u^x_{i+\kappa_x/2,j}, \Delta u^y_{i,j+\kappa_y/2}$ are derived by the 1D TVD limiter,

$$\overline{u}_{i,j+\kappa_y} \leq \overline{u}_{ij} + \Delta u^y_{i,j+\kappa_y/2},$$

$$\overline{u}_{ij} + \Delta u^x_{i+\kappa_x/2,j} \leq \overline{u}_{i+\kappa_x,j}.$$

Therefore,

$$\overline{u}^{\min}_{\kappa_x,\kappa_y} \leq \overline{u}_{i,j+\kappa_y} \leq u_{i+\kappa_x/2,j+\kappa_y/2} \leq \overline{u}_{i+\kappa_x,j} \leq \overline{u}^{\max}_{\kappa_x,\kappa_y}.$$

□

Proposition 4.3.2. *When* $\Delta u^x_{i+\kappa_x/2,j} > 0\ (\leq 0)$, $\Delta u^y_{i,j+\kappa_y/2} > 0\ (\leq 0)$ (i.e max(min) vertex)*, the left*(*right*) *inequality in the limiting condition* (4.3.1)

is satisfied.

Proof.

$$\overline{u}^{\min}_{\kappa_x,\kappa_y} \le \overline{u}_{ij} \le \overline{u}_{ij} + \Delta u^x_{i+\kappa_x/2,j} + \Delta u^y_{i,j+\kappa_y/2} = u_{i+\kappa_x/2,j+\kappa_y/2}.$$

□

Since vertex point values that are neither maximum nor minimum satisfy (4.3.1), it is enough to show the following holds.

$$\begin{aligned} u_{i+\kappa_x/2,j+\kappa_y/2} &\le \max(\overline{u}_{i,j}, \overline{u}_{i+\kappa_x,j}, \overline{u}_{i,j+\kappa_y}, \overline{u}_{i+\kappa_x,j+\kappa_y}) \\ &\text{when } \ u_{i+\kappa_x/2,j+\kappa_y/2} \ \text{ is a maximum vertex,} \end{aligned} \tag{4.3.5}$$

$$\begin{aligned} \min(\overline{u}_{i,j}, \overline{u}_{i+\kappa'_x,j}, \overline{u}_{i,j+\kappa'_y}, \overline{u}_{i+\kappa'_x,j+\kappa'_y}) &\le u_{i+\kappa'_x/2,j+\kappa'_y/2} \\ &\text{when } \ u_{i+\kappa'_x/2,j+\kappa'_y/2} \ \text{ is a minimum vertex.} \end{aligned} \tag{4.3.6}$$

Yoon et al. computed parameters α_x, α_y satisfying (4.3.5) and (4.3.6) in [34].

We now introduce the relieved maximum property, which is a simple modification of the original maximum property to control the contribution of the MLP parameters α_x and α_y requiring the MLP condition and reducing the dissipation near discontinuities. We construct the new MLP parameters from the relieved maximum property

$$\overline{u}_{ij} - c \le u_{i+\kappa_x/2,j+\kappa_y/2} \le \overline{u}_{ij} + c \tag{4.3.7}$$

with $c = \frac{1}{2}(\overline{u}^{\max}_{\kappa_x,\kappa_y} - \overline{u}^{\min}_{\kappa'_x,\kappa'_y})$. In order to determine the parameters α_x and α_y satisfying (4.3.7), we have the following inequality from (4.3.4) :

$$\overline{u}_{ij} - c \le \overline{u}_{ij} + \Delta u^x_{i+\kappa_x/2,j} + \Delta u^y_{i,j+\kappa_y/2} \le \overline{u}_{ij} + c. \tag{4.3.8}$$

Then we obtain

$$|(1 + r_{xy})\Delta u^x_{i+\kappa_x/2,j}| \le c \tag{4.3.9}$$

with $r_{xy} = \dfrac{\Delta u^y_{i,j+\kappa_y/2}}{\Delta u^x_{i+\kappa_x/2,j}}$. We now assume that $u_{i+\kappa_x,j+\kappa_y}$ is the maximum vertex, $r_{xy} > 0$ and $\Delta u^x_{i+\kappa_x/2,j} > 0$. By the definition of the deviation term, we have

$$\left|\frac{\kappa_x}{2}\phi(r_x)\Delta\overline{u}_{i-1/2,j}\right| \le \frac{c}{1+r_{xy}}$$

and

$$\max(0, \min(\alpha_x, \alpha_x r_x)) \le \frac{2c}{\kappa_x(1+r_{xy})\Delta\overline{u}_{i-1/2,j}}.$$

We may assume $r^L_x > 0$, so we have

$$\alpha_x \min(1, r^L_x) \le \left|\frac{2c}{\kappa_x(1+r_{xy})\Delta\overline{u}_{i-1/2,j}}\right|.$$

Thus we obtain

$$\begin{aligned}\alpha_x &\le \left|\frac{2c\max(1, 1/r^L_x)}{(1+r_{xy})\Delta\overline{u}_{i-1/2,j}}\right| \\ &= \left|\frac{2c\max(1, 1/r^L_x)r^L_x}{(1+r_{xy})\Delta\overline{u}_{i-1/2,j}r^L_x}\right| \\ &= \left|\frac{2c\max(1, r^L_x)}{(1+r_{xy})\Delta\overline{u}_{i+1/2,j}}\right|\end{aligned} \tag{4.3.10}$$

with $c = \frac{1}{2}(\overline{u}^{\max}_{\kappa_x,\kappa_y} - \overline{u}^{\min}_{\kappa'_x,\kappa'_y})$. In the case that $u_{i+\kappa_x/2,j+\kappa_y/2}$ is a minimum, by the symmetry of (4.3.9), it is easy to check that (4.3.10) is also derived. It is a simple observation that $u_{i-\kappa_x/2,j-\kappa_y/2}$ is a minimum if $u_{i+\kappa_x/2,j+\kappa_y/2}$ is a maximum facilitate vertex. Choosing the maximum α_x, finally, we have

$$\alpha_x = \left|\frac{(\overline{u}^{\max}_{\kappa_x,\kappa_y} - \overline{u}^{\min}_{-\kappa_x,-\kappa_y})\max(1, r^L_x)}{(1+r_{xy})\Delta\overline{u}_{i+1/2,j}}\right|.$$

To summarize the overall process, the formula used to implement CU-MLP are presented in the following Algorithm 6.

Algorithm 6 Evaluation the cell-interface values with MLP

$$\kappa_x = \text{sign}(\Delta u_{i-1/2,j}), \quad \kappa_y = \text{sign}(\Delta u_{i,j-1/2})$$
$$r_{xy} = \frac{\Delta \overline{u}^y_{i,j+\kappa_y/2}}{\Delta \overline{u}^x_{i+\kappa_x/2,j}} \approx \left| \frac{\overline{u}_{i,j+1} - \overline{u}_{i,j-1}}{\overline{u}_{i+1,j} - \overline{u}_{i-1,j}} \right|, \quad r_{yx} = \frac{1}{r_{xy}}$$

$$(\alpha_x)_{ij} = \begin{cases} \left| \dfrac{(u^{\max}_{\kappa_x,\kappa_y} - u^{\min}_{-\kappa_x,-\kappa_y}) \max(1, (r_x)_{ij})}{(1 + r_{xy}) \Delta \overline{u}_{i+1/2,j}} \right| & \text{if } (r_x)_{ij} > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4.3.11}$$

$$(\alpha_y)_{ij} = \begin{cases} \left| \dfrac{(u^{\max}_{\kappa_x,\kappa_y} - u^{\min}_{-\kappa_x,-\kappa_y}) \max(1, (r_y)_{ij})}{(1 + r_{yx}) \Delta \overline{u}_{i,j+1/2}} \right| & \text{if } (r_y)_{ij} > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4.3.12}$$

$$u^L_{i+1/2,j} = \overline{u}_{ij} + 0.5\phi^L((r_x)_{ij}) \triangle \overline{u}_{i-1/2,j}.$$
$$u^R_{i-1/2,j} = \overline{u}_{ij} - 0.5\phi^R((r^R_x)_{ij}) \triangle \overline{u}_{i+1/2,j}.$$
$$u^L_{i,j+1/2} = \overline{u}_{ij} + 0.5\phi^L((r_y)_{ij}) \triangle \overline{u}_{i,j-1/2}.$$
$$u^R_{i,j-1/2} = \overline{u}_{ij} - 0.5\phi^R((r^R_y)_{ij}) \triangle \overline{u}_{i,j+1/2}.$$

4.4 Numerical results

In this section, we present numerical results of the central-upwind scheme with modified MLP limiter. We compare the numerical performance of the proposed scheme. The numerical example starts with the solution of the 2-dimensional advection equation, followed by the solution of Burgers equation, and 2D Euler systems of equations with Riemann initial-value problems. First, we apply our schemes to the scalar 2D advection equation with proper initial conditions to test the propagation of arbitrary initial profiles containing jump discontinuities and corner points.

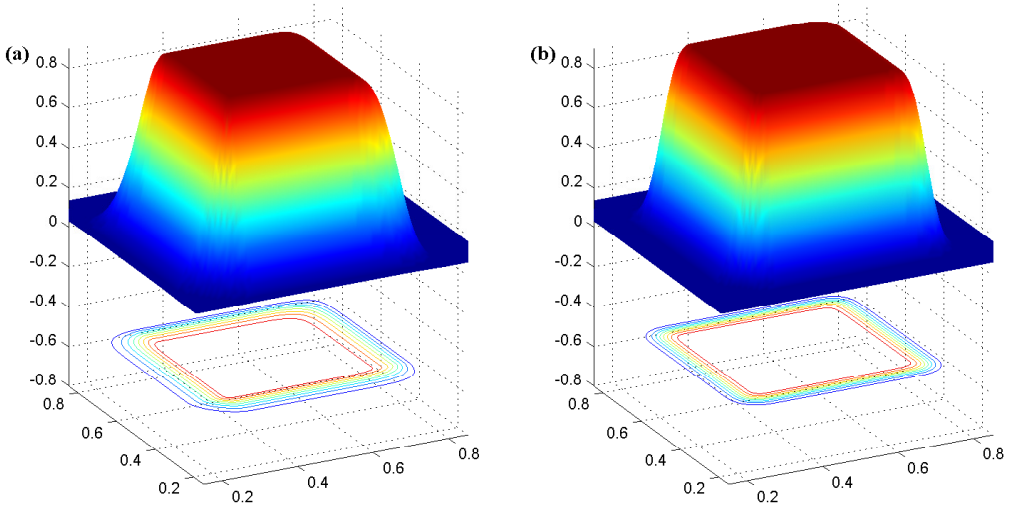


Figure 4.2: CU-KNP (left) and CU-MLP(right) with $\Delta x = \Delta y = 1/100$ at $t = 4$.

4.4.1 Linear advection equation

we apply the CU-MLP schemes to the scalar advection equation with a discontinuous initial condition to compare the behavior of the proposed schemes at discontinuities. We consider the following linear advection equation,

$$q_t + q_x + q_y = 0, \qquad (x, y) \in [0, 1] \times [0, 1], \tag{4.4.1}$$

with the initial condition,

$$q(x, y, 0) = q_0(x, y) = \begin{cases} 1 & (x, y) \in Q, \\ 0.1 & (x, y) \in Q^c \end{cases} \tag{4.4.2}$$

where $Q = [0.25, 0.75] \times [0.25, 0.75]$. We solve the equation (4.4.1) up to $t = 2$ to investigate the stability and the amount of smearing at discontinuities of the proposed scheme. The computational results are shown in figure 4.2. We can observe that discontinuities smear less for CU-MLP than for CU-KNP.

4.4.2 Burger's equation

We approximate solutions of the inviscid Burgers's equation

$$q_t + \left(\frac{q^2}{2}\right)_x + \left(\frac{q^2}{2}\right)_y = 0, \qquad (x, y) \in [0, 1] \times [0, 1], \tag{4.4.3}$$

with the initial condition

$$q(x, y, 0) = q_0(x, y) = \begin{cases} 1, & (x, y) \in Q \\ 0.1, & (x, y) \in Q^c \end{cases} \tag{4.4.4}$$

where $Q = [0.1, 0.6] \times [0.1, 0.6]$. Figure 4.3 shows the solutions at time t = 2. The CU-KNP schemes smear the shock slightly more than the proposed CU-MLP schemes.

4.4.3 2D Euler system - Four shocks

We consider numerical solutions of the 2D Riemann problems originally defined in [70]. This problem is solved on a square domain $[0, 1] \times [0, 1]$, with initial constant states in four quadrants. The square is divided into four quadrants by lines $x = 0.8$ and $y = 0.8$. We take the initial condition as constant states on each of four quadrants and compute this initial data up to time $t = 0.8$. The Riemann problems are defined by initial constant states in each quadrant:

$$(\rho, u, v, p) = \begin{cases} (1.5, 0, 0, 1.5) & \text{if } 0.8 \le x \le 1, 0.8 \le y \le 1, \\ (0.5323, 1.206, 0, 0.3) & \text{if } 0 \le x < 0.8, 0.8 \le y \le 1, \\ (0.138, 1.206, 1.206, 0.029) & \text{if } 0 \le x < 0.8, 0 \le y < 0.8, \\ (0.5323, 0, 1.206, 0.3) & \text{if } 0.8 < x \le 1, 0 \le y < 0.8. \end{cases}$$

We set the gas constant $\gamma = 1.4$. To the best of our knowledge, the exact solution has not been elucidated for this 2D problem. We compare the numerical performance of the CU-KNP scheme with the proposed schemes in figure 4.4. An examination of these results reveals that the proposed scheme

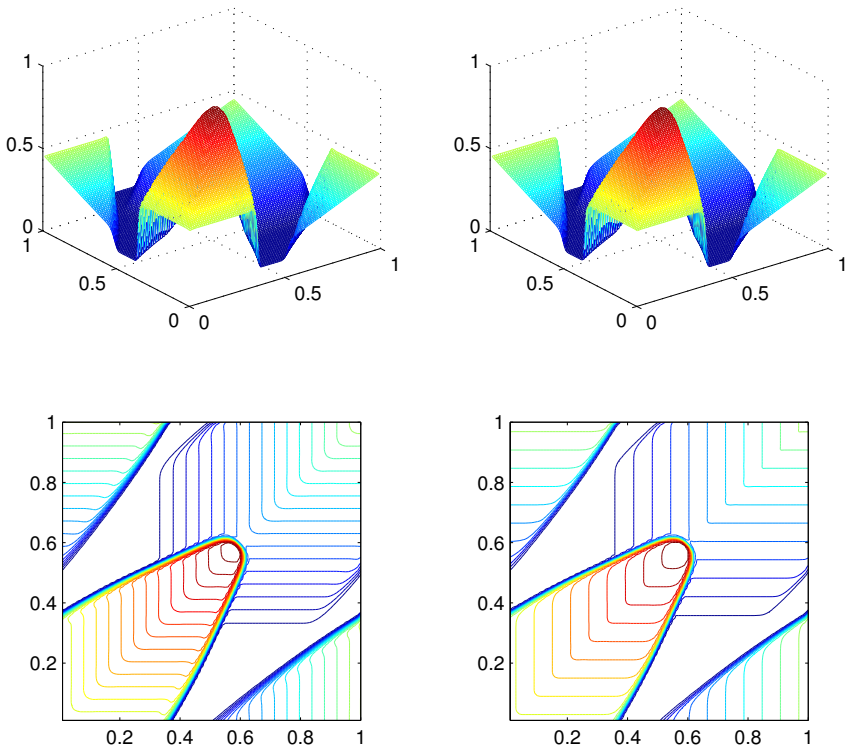


Figure 4.3: 2D Burgers CU-KNP (left) a nd CU-MLP(right) with $\Delta x = \Delta y = 1/100$ at $t = 2$.

exhibits a better resolution of the structure appearing in CU-KNP schemes.

4.4.4 2D Euler system - Rayleigh-Taylor instability

Taylor instability in two-dimensional incompressible fluids happens on an interface between two fluids of different densities when the lighter fluid is pushing the heavier fluid. Several experiments are reported extensively in the literature (e.g. [71] and [72]). This problem is solved on a computational

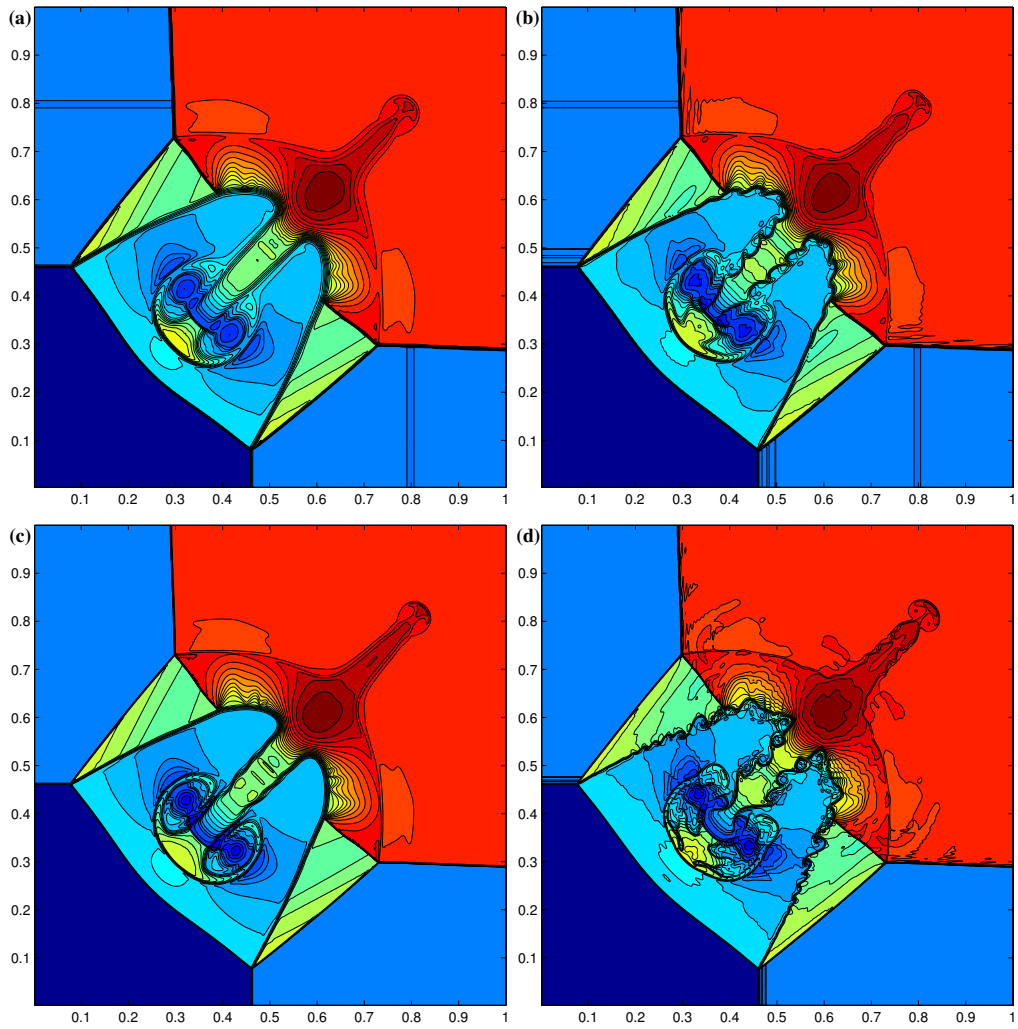


Figure 4.4: 2D Euler-four-shocks (a) CU-KNP and (b) CU-MLP with $\Delta x = \Delta y = 1/400$ and (c) CU-KNP and (d) CU-MLP with $\Delta x = \Delta y = 1/800$ at $t = 0.8$.

domain $[0, 0.25] \times [0, 1]$ with the following initial condition:

$$(\rho, u, v, p) = \begin{cases} (2, 0, -0.025c\cos(8\pi x), 2y + 1) & \text{if } 0 \le y < 0.5, \\ (1, 0, -0.025c\cos(8\pi x), y + \frac{3}{2}) & \text{if } 0.5 \le y < 1. \end{cases}$$

Here, $c = \sqrt{\gamma p/\rho}$ and $\gamma = 5/3$ are the sound speed and the ratio of specific heats respectively. Adding ρ and ρv to the right hand side of third and fourth

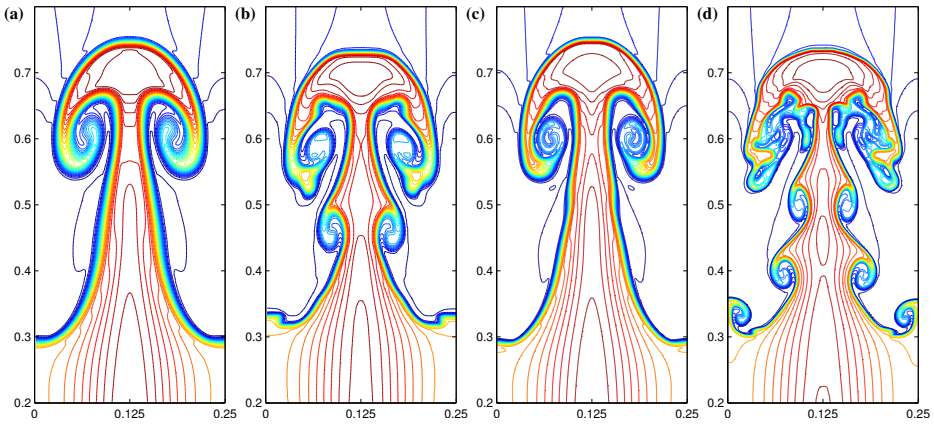


Figure 4.5: Density profiles of Rayleigh–Taylor instability (a) CU-KNP and (b) CU-MLP with $\Delta x = \Delta y = 1/240$ and (c) CU-KNP and (d) CU-MLP with $\Delta x = \Delta y = 1/480$ at $t = 1.95$

equation to have the gravitational effect, respectively. Reflective boundary conditions are applied for the left and right boundaries, and

$$(\rho, u, v, p) = \begin{cases} (2, 0, 0, 2.5) & \text{top boundary,} \\ (1, 0, 0, 1) & \text{bottom boundary.} \end{cases}$$

The simulation time is $t = 1.95$. From the density contours with $\Delta x = \Delta y = 1/480$ plotted in figure 4.5, it can be seen that the proposed scheme obtains more complex structures than the CU-KNP schemes.

4.4.5 2D Euler system - Double Mach reflection of a strong shock

The considered test case is the two dimensional double Mach reflection of a shock off an oblique surface, which is also a very popular test case for high resolution schemes [62]. The whole computational domain is $[0, 4] \times [0, 1]$ with equally spaced grid points. Initially, a rightmoving shock with Mach number 10 is located at the bottom of computational domain x-axis at $x = \frac{1}{6}$, inclined at a 60° angle with respect to the x-axis. Reflective boundary condition is applied along the bottom wall, and the top boundary of the

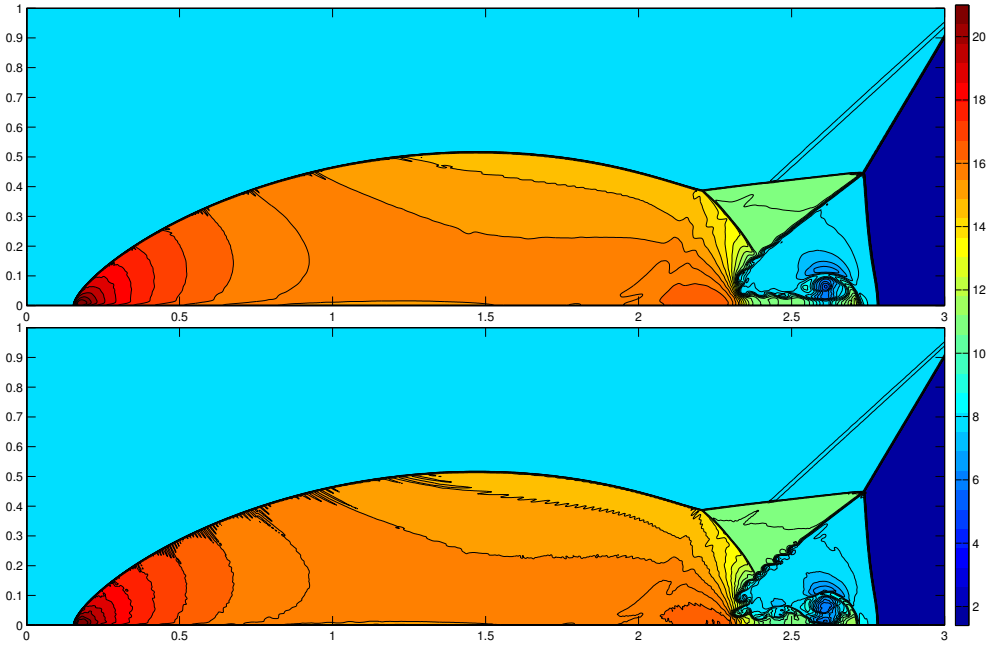


Figure 4.6: Density profiles of Double Mach reflection of a strong shock [62] at $t = 0.2$ with $\Delta x = \Delta y = 1/480$ (top : CU-KNP, bottom : CU-MLP).

problem is set to describe the exact motion of the Mach 10 shock. See [62] for a detailed description of this problem. We display the results in $[0, 3] \times [0, 1]$ as customary. Figure 4.6 shows the details at the Mach stem of the density variable at the final time at $t = 0.2$ with $\gamma = 1.4$ and $\Delta x = \Delta y = 1/480$. We can clearly see that the proposed scheme resolves the instabilities around the Mach stem.

Chapter 5

Conclusions

In this work, by applying expensive WENO or WENO-Z schemes only at points with maximum levels, and applying fast fixed stencil differentiations at other points, we saved a large amount of computation while obtaining results that are almost same as those of full grid WENO schemes. For example, in case of the double mach reflection problem, we can see from the Figure 3.11 that the value over quite a large portion of the computational domain remains constant. Over those constant regions, we spared the potential application of a large number of WENO schemes. The wavelet based adaptive WENO schemes are considerably faster than full grid schemes. A series of numerical simulations obtained by the proposed method and the full grid method are displayed with negligible differences with 6 or 7 levels of grid scales. This implies that very complicated vortex system such as Orszag-Tang could be resolved according to the physical scales.

Our adaptive algorithm is not restricted by application only with WENO schemes. In fact, it is not dependent on what high resolution scheme(HRS) is used for oscillation controls. It is easy to hybridize any other HRS with the adaptive algorithm.

Furthermore, central-upwind methods with modified multi-dimensional process (CU-MLP) have been introduced for the approximate solutions of two-dimensional hyperbolic conservation laws. Even though central-upwind schemes have the advantage of simplicity of the Riemann free with second

order accuracy, they have a little bigger dissipations compared to second order upwind schemes. To overcome this drawback we apply the MLP limiter to the central-upwind schemes, and this attempt results in excellent performances in all test cases. Compared to the original MLP parameter α, a modified MLP parameter for central-upwind methods resolves discontinuities sharply while keeping an essentially non-oscillatory performance. The improvement is attributed to the ability of proposed schemes to detect the complicated solution structures.

Bibliography

[1] S. K. Godunov, A Finite Difference Method for the Numerical Computation of Discontinuous Solutions of the Equations of Fluid Dynamics, Mat. Sb, Vol. 47, 271–290, (1959) in Russian.

[2] B. V. Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 101–136.

[3] G.-S. Jiang, C.-W. Shu, Efficient Implementation of Weighted ENO Schemes, J. Comput. Phys. 126 (1996) 202–228.

[4] C.-W. Shu, S. Osher, Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes, J. Comput. Phys. 77 (1988) 439–471.

[5] C.-W. Shu, S. Osher, Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes, II, J. Comput. Phys. 83 (1989) 32–78.

[6] A. K. Henrick, T. D. Aslam, J. M. Powers, Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points, J. Comput. Phys. 207 (2005) 542–567.

[7] R. Borges, M. Carmona, B. Costa, W. S. Don, An improved weighted essentially non-oscillatory scheme for hyperbolic conservaltion laws, J. Comput. Phys. 227 (2008) 3191–3211.

[8] Y. Ha, C. H. Kim, Y. J. Lee, J. Yoon, An improved weighted essentially non-oscillatory scheme with a new smoothness indicator, J. Comput. Phys. 232 (2013) 68–86.

[9] O. V. Vasilyev, C. Bowman, Second-Generation Wavelet Collocation Method for the Solution of Partial Differential Equations, Journal of Computational Physics 165 (2) (2000) 660 – 693.

[10] O. V. Vasilyev, Solving Multi-dimensional Evolution Problems with Localized Structures using Second Generation Wavelets, International Journal of Computational Fluid Dynamics 17 (2003) 151–168.

[11] K. Friedrichs, Symmetric hyperbolic linear differential equations, Comm. Pure Appl. Math. 7, 345–392 (1954).

[12] P. Lax, Weak solutions of nonlinear hyperbolic equations and their numerical computation, Comm. Pure Appl. Math. 7, 159–193 (1954).

[13] E. T. H. Nessyahu, Nonoscillatory central differencing for hyperbolic conservation laws, J. Comput. Phys. 87, 408–463 (1990).

[14] E. T. A. Kurganov, New High-Resolution Central Schemes for Nonlinear Conservation Laws and Convection–Diffusion Equations, em J. Comput. Phys. 160, 241–282 (2000).

[15] E. T. G. S. Jiang, Non-oscillatory central schemes for multidimensional hyperbolic conservation laws, SIAM J. Sci. Comput. 19, 1892–1917 (1998).

[16] G. P. A. Kurganov, S. Noelle, Semidiscrete central-upwind schemes for hyperbolic conservation laws and Hamilton-Jacobi equations, SIAM J. Sci. Comput. 23, 707–740 (2001).

[17] X. D. Liu, S. Osher, Nonoscillatory high order accurate self-similar maximum principle satisfying shock capturing schemes I, SIAM J. Numer. Anal. 33, 760–779 (1996).

[18] E. T. X.-D. Liu, Third order nonoscillatory central scheme for hyperbolic conservation laws, Numer. Math. 79, 397–425(1998).

[19] S. O. A. Harten, B. Engquist, Uniformly high order accurate essentially non-oscillatory schemes, J. Comput. Phys. 71, 231–303 (1987).

[20] T. C. X. D. Liu, S. Osher, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 15, 200–212 (1994).

[21] G. P. F. Bianco, G. Russo, High order central schemes for hyperbolic systems of conservation laws, SIAM J. Sci. Comput. 21, 294–322 (1999).

[22] G. P. D. Levy, G. Russo, Central WENO schemes for hyperbolic systems of conservation laws, Math. Model. Numer. Anal. 33, 547–571 (1999).

[23] G. P. D. Levy, G. Russo, Compact central WENO schemes for multidimensional conservation laws, SIAM J. Sci. Comput. 22, 656–672 (2000).

[24] J. Qiu, C. W. Shu, On the construction, comparison, and local characteristic decomposition for high-order central WENO schemes, J. Comput. Phys. 183, 187–209 (2002).

[25] G. R. D. Levy, G. Puppo, A third order central WENO scheme for 2D conservation laws, Appl. Numer. Math. 33, 415–421 (2000).

[26] G. P. D. Levy, G. Russo, A fourth-order central WENO scheme for multidimensional hyperbolic systems of conservation laws, SIAM J. Sci. Comput 24, 480–456 (2002).

[27] P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, J. Comput. Phys. 87, 171–200 (1990).

[28] P. van Ransbeeck, C. Hirsch, New upwind dissipation models with a multidimensional approach, AIAA paper 92–0436 (1992).

[29] D. Sidikover, Multidimensional upwinding and multigrid, AIAA paper 95–1759 (1995).

[30] K. H. Kim, C. Kim, Accurate, efficient and monotonic numerical methods for multi-dimensional compressible flows Part II: Multi-dimensional limiting process, J. Comput. Phys. 208 (2) (2005) 570–615.

[31] P. Gerlinger, Multi-dimensional limiting for high-order schemes including turbulence and combustion, J. Comput. Phys. 231, 2199–2228 (2012).

[32] D. H. L. H. M. Kang, K. H. Kim, A new approach of a limiting process for multi-dimesnional flows, J. Comput. Phys. 229, 7102–7128 (2010).

[33] K. H. K. S. Kim, S. Lee, Wavenumber-extended high-order oscillation control finite volume schemes for multi-dimensional aeroacoustic simulations, J. Comput. Phys. 227, 4089–4122 (2008).

[34] K. H. K. S. H. Yoon, C. Kim, Multi-dimensional limiting process for three-dimensional flow physics analyses, J. Comput. Phys 227, 6001–6043 (2008).

[35] Randall J. Leveque, Finite Volume Methods for Hyperbolic Problems, 2002.

[36] D. Ryu, T. Jones, Numerical magnetohydrodynamics in astrophysics: Algorithm and tests for one-dimensional flow, Astrophysical Journal 442 (1) (1995) 228–258.

[37] G. Tóth, The $\nabla \cdot$ B=0 Constraint in Shock-Capturing Magnetohydrodynamics Codes, Journal of Computational Physics 161 (2) (2000) 605 – 652.

[38] A. Dedner, F. Kemm, D. Kröner, C.-D. Munz, T. Schnitzer, M. Wesenberg, Hyperbolic Divergence Cleaning for the MHD Equations, Journal of Computational Physics 175 (2) (2002) 645 – 673.

[39] C. R. Evans, J. F. Hawley, Simulation of magnetohydrodynamic flows - A constrained transport method, Astrophysical Journal 332 (1988) 659–677.

[40] C.-D. Munz, P. Omnes, R. Schneider, E. Sonnendrücker, U. Voß, Divergence Correction Techniques for Maxwell Solvers Based on a Hyperbolic Model, Journal of Computational Physics 161 (2) (2000) 484 – 511.

[41] J. U. Brackbill, D. C. Barnes, The effect of nonzero product of magnetic gradient and B on the numerical solution of the magnetohydrodynamic equations, Journal of Computational Physics 35 (1980) 426–430.

[42] M. Castro, B. Costa, W. S. Don, High Order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conseravtion laws, J. Comput. Phys. 230 (2011) 1766–1792.

[43] S. Mallat, Multiresolution approximations and wavelet orthonormal bases of $L^2(\mathbb{R})$, Trans. Amer. Math. Soc. 315 (1989) 69–87.

[44] Yves Meyer, Wavelets and Operators, 1992.

[45] Mallat, S., A Wavelet Tour of Signal Processing, 2009.

[46] G. Deslauriers, S. Dubuc, Symmetric iterative interpolation processes, Constr. Approx. 5 (1989) 49–68.

[47] S. Dubuc, Interpolation through an iterative scheme, Journal of Mathematical Analysis and Applications 114 (1) (1986) 185 – 204.

[48] O. V. Vasilyev, N. K.-R. Kevlahan, An adaptive multilevel wavelet collocation method for elliptic problems, Journal of Computational Physics 206 (2) (2005) 412 – 431.

[49] W. Sweldens, The lifting scheme: A custom-design construction of biorthogonal wavelets, Appl. Comput. Harmon. Anal. 315 (1996) 186–200.

[50] W. Sweldens, The lifting scheme: A construction of second generation wavelets, SIAM J. Math. Anal. 29 (2) (1998) 511–546.

[51] D. L. Donoho, Interpolating wavelet transforms, Tech. rep. (1992).

[52] J. Regele, O. Vasilyev, An adaptive wavelet-collocation method for shock computations, International Journal of Computational Fluid Dynamics 23 (7) (2009) 503–518.

[53] S. Bochner, K. Chandrasekharan, Fourier Transforms, Princeton University Press, 1949.

[54] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge-Kutta schemes, Mathematics of Computation 67 (221) (1998) 73–85.

[55] M. O. Domingues, S. M. Gomes, O. Roussel, K. Schneider, An adaptive multiresolution scheme with local time stepping for evolutionary PDEs , Journal of Computational Physics 227 (8) (2008) 3758 – 3780.

[56] M. Dumbser, O. Zanotti, A. Hidago, D. S. Balsara, ADER-WENO finite volume schemes with space-time adaptive mesh refinement, J. Comput. Phys. 248 (2013) 257–286.

[57] E. Pärt-Enander, B. Sjögreen, Conservative and non-conservative interpolation between overlapping grids for finite volume solutions of hyperbolic problems, Computers and Fluids 23 (3) (1994) 551 – 574.

[58] K. Sebastian, C.-W. Shu, Multidomain WENO Finite Difference Method with Interpolation at Subdomain Interfaces, Journal of Scientific Computing 19 (1-3) (2003) 405–438.

[59] H. Li, S. Do, M. Kang, A wavelet-based adaptive WENO algorithm for Euler equations, Computers and Fluids 123 (2015) 10 – 22.

[60] P. Lax, X.-D. Liu, Solution of two-dimensional Riemann problems of gas dynamics by positive schemes, SIAM Journal on Scientific Computing 19 (2) (1998) 319–340.

[61] C.-W. Shu, Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws, ICASE Report (97-65).

[62] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, Journal of Computational Physics 54 (1) (1984) 115 – 173.

[63] Z. Xu, C.-W. Shu, Anti-diffusive flux corrections for high order finite difference WENO schemes, Journal of Computational Physics 205 (2) (2005) 458 – 485.

[64] S. A. Orszag, C.-M. Tang, Small-scale structure of two-dimensional magnetohydrodynamic turbulence, Journal of Fluid Mechanics 90 (1979) 129–143.

[65] A. L. Zachary, A. Malagoli, P. Colella, A Higher-Order Godunov Method for Multidimensional Ideal Magnetohydrodynamics, SIAM Journal on Scientific Computing 15 (2) (1994) 263–284.

[66] W. Dai, P. R. Woodward, A Simple Finite Difference Scheme for Multidimensional Magnetohydrodynamical Equations, Journal of Computational Physics 142 (2) (1998) 331 – 369.

[67] D. Ryu, F. Miniati, T. W. Jones, A. Frank, A Divergence-free Upwind Code for Multidimensional Magnetohydrodynamic Flows, The Astrophysical Journal 509 (1) (1998) 244.

[68] D. S. Balsara, D. S. Spicer, A Staggered Mesh Algorithm Using High Order Godunov Fluxes to Ensure Solenoidal Magnetic Fields in Magnetohydrodynamic Simulations, Journal of Computational Physics 149 (2) (1999) 270 – 292.

[69] P. K. Sweby, High resolution schemes using flux limiters for hyperbolic conservation laws, SIAM J. Numer. Anal. 21 (5) (1984) 995–1011.

[70] J. C. C.W. Schulz-Rinne, H. Glaz, Numerical Solution of the Riemann Problem for Two-Dimensional Gas Dynamics, SIAM J. Sci. Comput., Vol. 14, No. 6, 1394–1414 (1993).

[71] J. G. J. Glimm, D. Tan, The dynamics of bubble growth for Rayleigh–Taylor unstable interfaces, Physics of Fluids 31, 447–465 (1988).

[72] A. D. Y. Young, H. Tufo, R. Rosner, On the miscible Rayleigh–Taylor instability: two and three dimension, Journal of Fluid Mechanics 447, 377–408 (2001).

국문초록

본 학위 논문에서, 우리는 Euler 방정식과 이상 자기유체 방정식과 같은 쌍곡 보존 법칙들을 풀어내는 효율적이고 고차 정확도를 갖는 수치 기법들에 대해 연구를 진행하였다.

첫 번째 방법은 웨이블렛 기반의 적응형 WENO(편중된 본질적 비진동) 방법이다. 유한 차분 WENO 방법은 강건하고 안정적이며, 높은 정확도를 갖는 수치기법으로 쌍곡 보존 법칙뿐만 아니라, 많은 편미분방정식을 푸는데 이용되어 왔다. 그러나 특성화 분해와 WENO 보간방법의 비선형적 가중치 계산과 같은 부분이 알고리즘의 계산양을 증가시킨다. 이에 우리는 웨이블렛 분석을 통하여, 특이점을 감지하고 더 나아가 수치적 미분 방법과 그리드의 해상도에 계층적 차이를 두는 방법을 제안한다.

두 번째로, 변형된 다차원 제한 법을 CU(Central-Upwind) 방법에 적용하였다. CU 방법은 리만 해결자와 특성화 분해가 필요하지 않아 효율적이지만, 접촉불연속면을 부드럽게 만드는 문제점이 있다. 이러한 단점을 극복하기 위해, 우리는 다차원 제한 법을 적용하였다.

주요어휘: 유한 차분에 기반한 WENO 방법, 소파 분석, 격자의 차용방법, 오일러 방정식, 이상적 자기유체역학 방정식, 충격파 감지.
학번: 2010–20244